Tools for dynamic model development

by

Spencer Daniel Schaber

B.Ch.E., University of Minnesota (2008)
M.S.C.E.P., Massachusetts Institute of Technology (2011)

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Author . . . .
Department of Chemical Engineering
April 18, 2014

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

Accepted by . . . .
Patrick S. Doyle
Chairman, Department Committee on Graduate Theses

Tools for dynamic model development

by

Spencer Daniel Schaber

Submitted to the Department of Chemical Engineering
on April 18, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

For this thesis, several tools for dynamic model development were developed and analyzed. Dynamic models can be used to simulate and optimize the behavior of a great number of natural and engineered systems, from the movement of celestial bodies to projectile motion to biological and chemical reaction networks. This thesis focuses on applications in chemical kinetic systems. Ordinary differential equations (ODEs) are sufficient to model many dynamic systems, such as those listed above. Differential-algebraic equations (DAEs) can be used to model any ODE system and can also contain algebraic equations, such as those for chemical equilibrium. Software was developed for global dynamic optimization, convergence order was analyzed for the underlying global dynamic optimization methods, and methods were developed to design, execute, and analyze time-varying experiments for parameter estimation and chemical kinetic model discrimination in microreactors. The global dynamic optimization and convergence order analysis thereof apply to systems modeled by ODEs; the experimental design work applies to systems modeled by DAEs.

When optimizing systems with dynamic models embedded, especially in chemical engineering problems, there are often multiple suboptimal *local* optima, so local optimization methods frequently fail to find the true (global) optimum. Rigorous global dynamic optimization methods have been developed for the past decade or so. At the outset of this thesis, it was possible to optimize systems with up to about five decision variables and five state variables, but larger and more realistic systems were too computationally intensive. The software package developed herein, called dGDOpt, for deterministic Global Dynamic Optimizer, was able to solve problems with up to nine parameters with five state variables in one case and a single parameter with up to 41 state variables in another case. The improved computational efficiency of the software is due to improved methods developed by previous workers for computing interval bounds and convex relaxations of the solutions of parametric ODEs as well as improved branch-and-bound heuristics developed in the present work.

The convergence order and prefactor were analyzed for some of the bounding and relaxation methods implemented in dGDOpt. In the dGDOpt software, we observed that the empirical convergence order for two different methods often differed, even though we suspected that both had the same analytical convergence order. In this thesis, it is proven that the bounds on the solutions of nonlinear ODEs converge linearly and the relaxations of the solutions of nonlinear ODEs converge quadratically for both methods. It is also proven that the convergence prefactor for an improved relaxation method can decrease over time, whereas the convergence prefactor for an earlier relaxation method can only increase over time, with worst-case exponential dependence on time. That is, the improved bounding method can actually shed conservatism from the relaxations as time goes on, whereas the initial method can only gain conservatism with time. Finally, it is shown how the time dependence of the bounds and relaxations explains the difference in empirical convergence order between the two relaxation methods.

Finally, a dynamic model for a microreactor system was used to design, execute, and analyze experiments in order to discriminate between models and identify the best parameters with less experimental time and material usage. From a pool of five candidate chemical kinetic models, a single best model was found and optimal chemical kinetic parameters were obtained for that model.

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

To my family and friends

Acknowledgments

> An expert is one who has made all the mistakes that can be made ...
>
> Niels Bohr

First, I would like to thank Prof. Paul Barton for helping to educate me as a whole individual, as he likes to say. He relentlessly pushed me to express myself precisely and this thesis would not have been possible without his numerous substantial questions and contributions. He also encouraged me to to present my work at international conferences, with all of the accompanying challenges and opportunities for growth.

I would like to thank my thesis committee. Prof. Richard Braatz pushed me to articulate the biggest challenges in the work of this thesis. Prof. Klavs Jensen gave me the privilege of working in his experimental lab to put my numerical methods into practice. Prof. Bill Green encouraged me to apply my numerical methods to the largest, most industrially-relevant problems possible.

I would like to thank the members of the Jensen lab, especially Brandon Reizman and Stephen Born, for welcoming me into their lab and helping me set up microreactor experiments.

The members of the Barton lab, especially Joe Scott, Geoff Oxberry, Matt Stuber, Achim Wechsung, Kai Höffner, Kamil Khan, Stuart Harwood, Siva Ramaswamy, and Ali Sahlodin, contributed to a fun working environment and helped me to understand many of the nuances of optimization, dynamic simulation, McCormick analysis, sensitivity analysis, programming, and more.

I would like to thank my friends from MIT ChemE, especially Christy, Mike, Achim, Andy, and Rachel for lots of refreshing lunches, dinners, parties, and adventures together.

Thanks to the members of the MIT Cycling Club—too many amazing people to name—with whom I had the privilege to explore the greater Boston area on group rides, race with a very strong team, and compete at national championships.

Thanks to my friends from home, especially Tim Bettger and Tim Usset, for helping me cement my values while we were in Minneapolis and keeping in touch throughout this degree.

Thanks to my teachers from the Mounds View school district and La Châtaigneraie, especially Graham Wright and Dan Butler, for conveying enthusiasm for math and science and demonstrating the value of discipline and hard work.

Thanks to Professors Curt Frank and Claude Cohen for mentoring me at your research labs during summer NSF REUs.

Thanks to all of my professors at the University of Minnesota, especially Ed Cussler, Alon McCormick, L. E. “Skip” Scriven, Raúl Caretta, Eray Aydil, and Frank Bates, as well as my graduate student mentor, Ben Richter.

Thanks to Mom, Dad, Andrew, and my grandparents. Mom and Dad, you showed me from an early age the value of hard work and achievement and you have become great friends and mentors.

Ali, thank you for all of the ways you cared for me during all of the fun and challenging stages of this degree.

Finally, I would like to acknowledge Novartis Pharmaceuticals for funding this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Dynamic models are often formulated as ordinary differential equations (ODEs) or differential-algebraic equations (DAEs). An ODE is a special case of a DAE that is theoretically and computationally easier to work with, while still being broadly applicable. Chemically reacting mixtures, vehicle dynamics, and a large variety of other systems can be modeled using ODEs.

We seek to optimize systems modeled by ODEs. Two examples addressed in this thesis are: (i) choosing the temperature profile along the length of a chemical reactor to maximize the concentration of a product and (ii) minimizing the difference between experimental data and model predictions by varying the model parameters. These types of optimizations are called *dynamic optimization* or *open-loop optimal control* problems. Another example not specifically addressed in this thesis would be choosing a flight path to minimize fuel consumption subject to constraints imposed by the airframe, flight dynamics of the aircraft, and government regulations.

1.1 Dynamic optimization methods

Broadly, dynamic optimization can be approached in two ways: so-called *direct* and *indirect* methods. Indirect methods derive optimality conditions for the original optimal control problem before discretizing these conditions, whereas direct methods use some form of discretization aimed at approximating the infinite-dimensional problem with a finite-

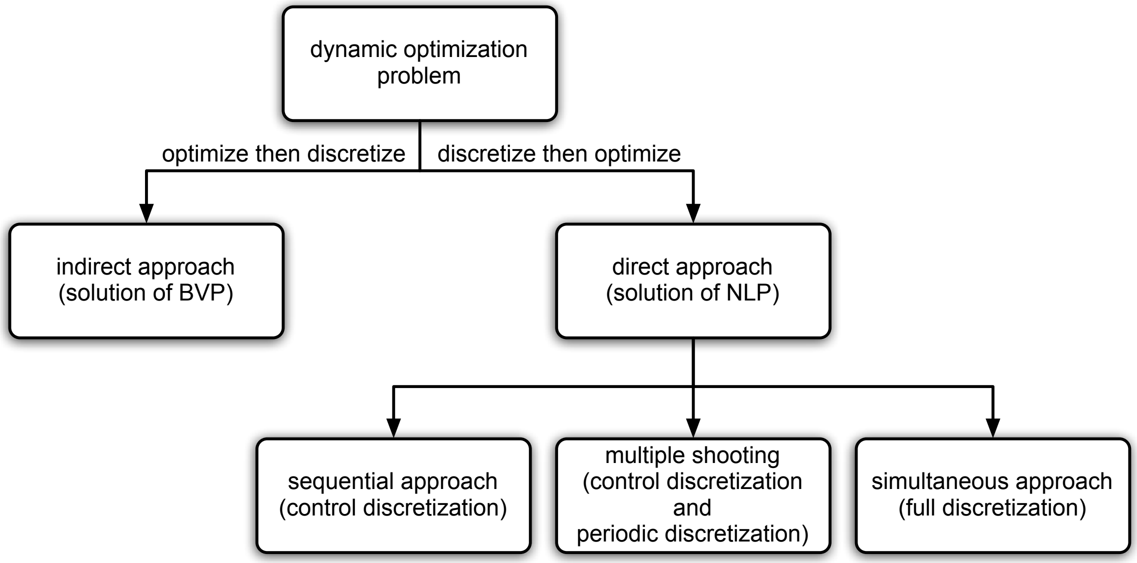

dynamic optimization problem
optimize then discretize
discretize then optimize
indirect approach (solution of BVP)
direct approach (solution of NLP)
sequential approach (control discretization)
multiple shooting (control discretization and periodic discretization)
simultaneous approach (full discretization)

Figure 1-1: Overview of dynamic optimization approaches, adapted from [28].

dimensional one. In this section, we discuss the existing indirect and direct methods for dynamic optimization. For a graphical overview of these methods, see Figure 1-1.

The two most common indirect methods are the *Hamilton-Jacobi-Bellman (HJB) equation* approach [18] and the *Pontryagin Maximum Principle (PMP)* [31, 152] also known as the Pontryagin *Minimum* Principle. The HJB equation approach is the continuous-time analog of dynamic programming [18], which can be used to solve discrete-time closed-loop optimal control problems. The HJB equation approach yields a partial differential equation (PDE) with an embedded minimization; the states and time are the independent variables of the PDE. By solving the PDE over the combined (time, state) space with an embedded minimization over the control space, the solution of this HJB PDE gives the optimal closed-loop control action for any value of the state and time. If the embedded minimization of the HJB equation can be solved to guaranteed global optimality, it yields a globally optimal solution of the optimal control problem. For convex problems, such as those with linear ODE models and quadratic costs, this is attainable, but for arbitrary nonconvex problems it amounts to embedding an NP-hard optimization problem within the solution of a PDE. If the state space has infinite cardinality (such as any problem where a state variable can take any real value in some range) and there is no analytical solution to the HJB PDE, then it becomes necessary to discretize the PDE in the state space and solve it at a finite number of values of the states. This is prohibitive in numerical implementations of the HJB approach with large numbers of state variables due to the curse of dimensionality inherent in solving a PDE with $n_x + 1$ independent variables, where n_x is the number of state variables. See [19, 20, 25, 30, 40, 78, 197] for additional information on the HJB. The PMP leads to a boundary-value problem that can be solved numerically [26, 40, 209]. In general, the PMP is a necessary but not sufficient condition for optimality. If it can be guaranteed that all possible solutions meeting the necessary conditions of the PMP can be found, then the best solution among those can be selected. However, in general this is very difficult to implement numerically.

Direct methods discretize the problem equations partially or fully. In the *partial discretization* approach also known as the *sequential* or *control vector parameterization* (CVP) approach, control functions are discretized into a vector of real-valued parameters whereas the states are evaluated as functions of these parameters by numerical integration of ODEs or DAEs. In *full discretization* also known as the *simultaneous discretization* approach, all of the control and state variables are discretized in time, yielding a nonlinear programming (NLP) problem with a large number of variables and constraints, as in [27, 29, 45, 91, 119]. The simultaneous approach has been attempted for global dynamic optimization, but due to the worst-case exponential running time of global NLP solvers and the very large number of optimization variables in the simultaneous approach, it can perform very badly [51, 67].

Here our focus is on direct methods, particularly partial discretization, which is also associated with the keywords *control vector parameterization* and *(single) shooting.* The infinite-dimensional problem of finding the control *function* minimizing some objective functional is reduced to a finite-dimensional problem by discretizing the control functions. Common choices of discretizations include piecewise constant, piecewise linear, and orthogonal polynomials such as Legendre polynomials. Within the piecewise control discretizations, the time discretization can be uniform or nonuniform, fixed or variable, and continuity of the control functions can be enforced or not enforced. See Figure 1-2 for a few examples of control discretization schemes. By parameterizing the controls into a vector of real parameters, optimal control problems and parameter optimization problems can be solved in the

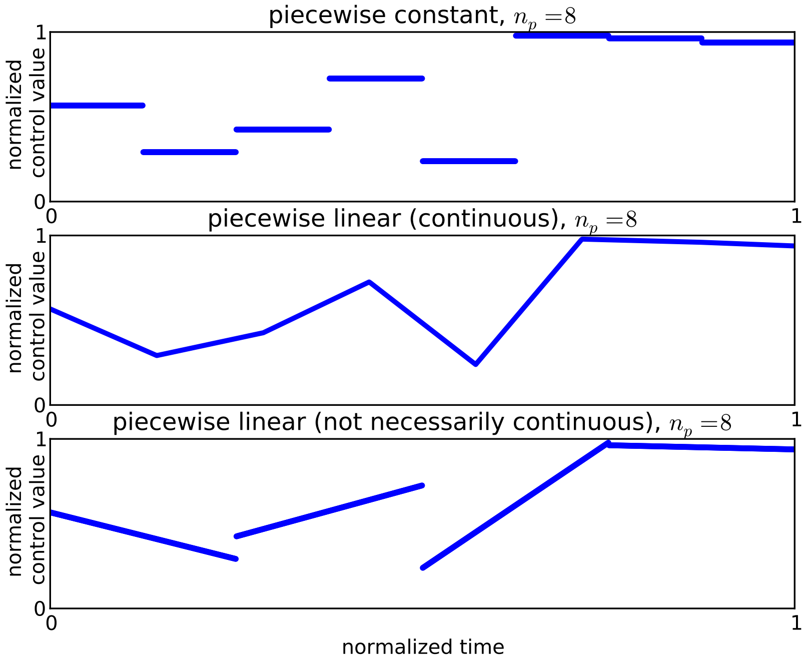


Figure 1-2: Control functions can be discretized in many ways. Each control discretization method above uses 8 control parameters.

same framework. See Figure 1-3 for an overview of the major steps in dynamic optimization based on the sequential approach. For further review of optimal control methods, see [26, 40, 197].

One more class of solution methods for optimal control methods is *multiple shooting*. Multiple shooting behaves something like a hybrid between single shooting, mentioned in the previous paragraph, and simultaneous or full discretization. The original optimal control problem is broken into several shooting problems, each with a portion of the original time horizon, and constraints are added to ensure that the states are continuous where the different time horizons intersect. This method yields NLPs intermediate in size between single shooting and full discretization approaches. The NLPs tend to be better conditioned than those arising from single shooting.

Chemical engineering dynamic optimization problems frequently have multiple suboptimal local minima. Luus and coworkers optimized a catalyst blend in a tubular reactor. With ten decisions, twenty-five local minima were found [121]. Singer et al. mentioned that when optimizing three chemical kinetic parameters for a seven-reaction, six-species

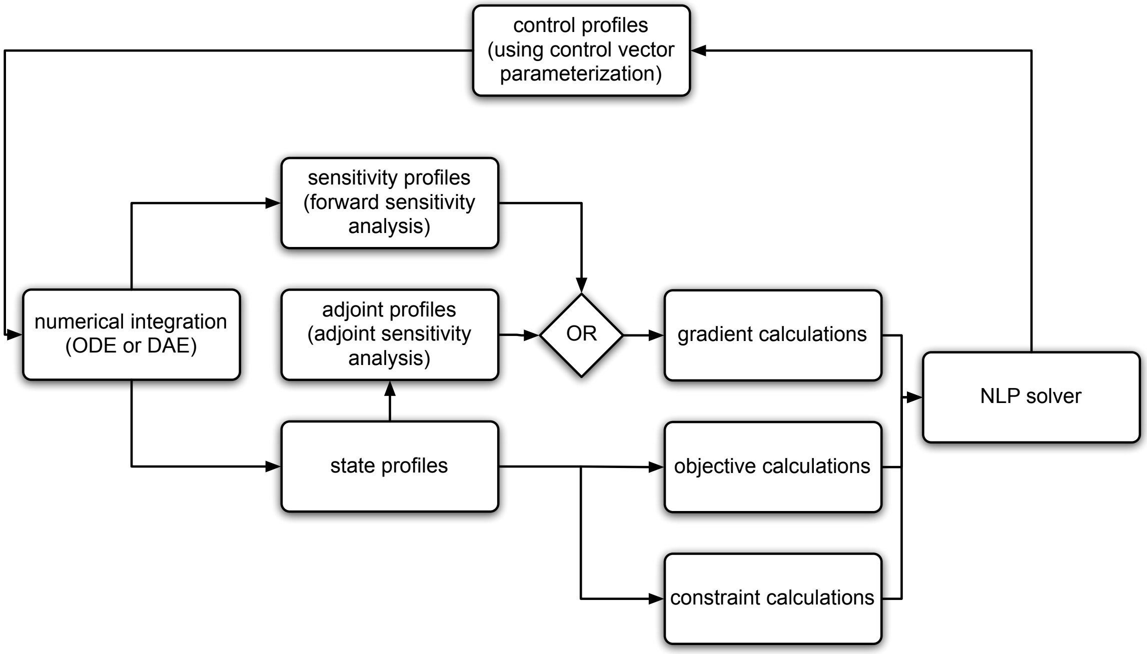

control profiles (using control vector parameterization)
sensitivity profiles (forward sensitivity analysis)
numerical integration (ODE or DAE)
adjoint profiles (adjoint sensitivity analysis)
OR
gradient calculations
NLP solver
state profiles
objective calculations
constraint calculations

Figure 1-3: Overview of the major steps of the sequential approach to dynamic optimization, adapted from [141].

system, hundreds of local minima were found [188]. Whereas most optimization software only finds local minima, we focus on methods that theoretically guarantee finding the best possible solution within a finite numerical tolerance. Software such as BARON [161] has been highly successful at solving nonconvex NLPs to guaranteed optimality; we seek to extend the success of global optimization methods to dynamic optimization problems.

1.2 Global optimization methods

Within global optimization, there are both stochastic and deterministic methods. Stochastic methods, such as simulated annealing [97], differential evolution [192], and others [14, 131, 156] have weak guarantees of convergence. Deterministic methods can have much stronger theoretical guarantees of convergence. In contrast to stochastic methods, deterministic methods, if properly designed, can theoretically guarantee convergence to within some $\varepsilon > 0$ tolerance in finite time. For an overview of global optimization applications and methods, see [143].

1.2.1 Branch-and-bound

The optimization methods herein are based on spatial branch-and-bound (B&B), a standard method used for global optimization of nonlinear programs (NLPs). Since any maximization problem can be trivially reformulated as a minimization problem, we consider only minimization problems. The basic idea of spatial B&B is to break a difficult (nonconvex) optimization problem into easier (convex) subproblems. The global optimum of a convex NLP can be obtained in polynomial time using a wide variety of optimization algorithms [24]. An optimization problem is *convex* if it has a convex feasible set and a convex objective function.

Definition 1.2.1 (Convex set)**.** A set $C \subset \mathbb{R}^n$ is *convex* if for every $\mathbf{x}, \mathbf{y} \in C$, the points

$$\mathbf{z} \equiv \lambda\mathbf{x} + (1-\lambda)\mathbf{y}, \quad \forall\lambda \in [0,1]$$

are also in the set C. In other words, C is convex if and only if every point on the line segment connecting any pair of points in C is also in C.

Definition 1.2.2 (Convex function)**.** Let $C \subset \mathbb{R}^n$ be a nonempty, convex set. A function $f : C \to \mathbb{R}$ is convex if it satisfies

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{y}), \quad \forall(\mathbf{x}, \mathbf{y}, \lambda) \in C \times C \times (0,1).$$

For a vector-valued function, the inequality must hold componentwise.

Definition 1.2.3 (Convex relaxation)**.** Given a nonempty convex $C \subset \mathbb{R}^n$ and function $f : C \to \mathbb{R}$, a function $u : P \to \mathbb{R}$ is a convex relaxation of f on C if u is convex and satisfies

$$u(\mathbf{x}) \leq f(\mathbf{x}), \quad \forall\mathbf{x} \in C.$$

Consider the NLP

$$\begin{aligned} &\min_{\mathbf{p}\in P} g(\mathbf{p}) \\ &\text{s.t. } \mathbf{h}(\mathbf{p}) \leq \mathbf{0}, \end{aligned} \tag{1.1}$$

where $P \subset \mathbb{R}^{n_p}$ is an n_p-dimensional interval and $g : P \to \mathbb{R}$ and $\mathbf{h} : P \to \mathbb{R}^{n_h}$ are continuous on P. To solve (1.1) to guaranteed global optimality, spatial B&B considers subproblems in which the feasible set is restricted to an interval $P^\ell \subset P$:

$$\begin{aligned} &\min_{\mathbf{p} \in P^\ell} g(\mathbf{p}) \\ &\text{s.t. } \mathbf{h}(\mathbf{p}) \le \mathbf{0}. \end{aligned} \tag{1.2}$$

To apply spatial B&B, we need a method of computing guaranteed lower- and upper-bounds for (1.2) for any $P^\ell \subset P$. Any feasible point in the optimization problem is a valid upper bound. Obtaining a rigorous lower bound is the difficult step. In this work, we obtain a lower bound on (1.2) by solving the relaxed optimization problem:

$$\begin{aligned} &\min_{\mathbf{p} \in P^\ell} g^{cv}(\mathbf{p}) \\ &\text{s.t. } \mathbf{h}^{cv}(\mathbf{p}) \le \mathbf{0}, \end{aligned} \tag{1.3}$$

where g^{cv} is a convex relaxations of g and $\mathbf{h}^{cv}$ is a convex relaxation of $\mathbf{h}$. Problem (1.3) is a convex optimization problem, so it can be solved to global optimality with standard NLP solvers. Since (1.3) is a relaxation of (1.2), the solution of (1.3) gives a lower bound on the solution of (1.2).

Alternatively, affine relaxations can be constructed to g and $\mathbf{h}$ and the resulting problem can be solved using a linear programming (LP) solver. The latter approach is more rigorous in the case that g^{cv} or $\mathbf{h}^{cv}$ cannot be guaranteed to be twice continuously differentiable. Such a LP relaxation can be readily constructed from the functions participating in (1.3) as long as subgradients are available. Such subgradients can be readily computed for McCormick relaxations [125] on a computer using the library MC++, which is the successor to libMC [130]. Since (1.3) is a relaxation of (1.2), it gives a lower bound on the optimal solution of (1.2), which is exactly what we need. In §2.1.1, we describe the generation of the convex relaxations g^{cv} and $\mathbf{h}^{cv}$ for dynamic optimization problems. See [84, 108, 143] for additional background on B&B.

1.2.2 Domain reduction

Domain reduction [129, 159, 160, 195] (or range reduction) techniques sometimes enable eliminating subsets of the optimization search space by guaranteeing that those subsets of the search space are guaranteed to be either suboptimal or infeasible. Domain reduction has helped make many more global optimization problem instances tracable. For branch-and-bound with the addition of domain reduction, the term *branch-and-reduce* has been coined [160]. For details of the domain reduction techniques used in this thesis, see §2.1.2.

1.2.3 Factorable function

The methods used for global optimization in this thesis rely on the participating functions being factorable [125, 135, 170]. Any function that can be represented finitely on a computer is factorable, including those with if statements and for loops. For our purposes, it will be sufficient for a function to be decomposable into a recursive sequence of operations, each of which is either addition, multiplication, or a univariate function from a given library of univariate functions.

1.2.4 Interval arithmetic

Given an interval $\widehat{P} \equiv \{\mathbf{p} \in \mathbb{R}^{n_p} : \mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U\}$ contained in the domain P of a factorable function f, interval arithmetic [134–136] can be used to generate a rigorous enclosure for the image $f(\widehat{P})$ of that input interval $\widehat{P}$ under f. Several libraries are available to calculate such enclosures, including INTLAB [158] for Matlab and PROFIL/BIAS [98], FILIB++ [111], and the BOOST interval arithmetic library [38] for C++. Once the factorable function is (automatically) decomposed into a sequence of individual operations, intervals are propagated through each of those constituent operations, finally yielding an interval guaranteed to enclose the image of the input interval for the overall function.

1.2.5 McCormick relaxations

The McCormick relaxation technique [125] allows point evaluation of a convex underestimator for any factorable function over any interval $\widehat{P}$ contained in the domain P of the function. Analogously to the procedure for interval arithmetic, McCormick [125] introduced rules for addition, multiplication, and univariate composition. When these rules are applied for each factor in the factored representation of the function, the end result is a procedure for point evaluation of a pair of convex and concave functions that underestimate and overestimate, respectively, the original function. As mentioned earlier, the libraries MC++ (`http://www.imperial.ac.uk/AP/faces/pages/read/Research.jsp?person=b.chachuat`) and libMC [130] can be used to compute such relaxations and their subgradients.

1.2.6 αBB relaxations

The αBB relaxation technique [1, 2] generates a convex relaxation of a function f by overpowering any nonconvexity in a function with a large convex term, yielding a relaxation of the form $f^{cv} : \mathbf{x} \mapsto f(\mathbf{x}) + \alpha \sum_{i=1}^{n_x} (x_i - x_i^L)(x_i - x_i^U)$. A large number of strategies have been demonstrated to calculate a sufficiently large value of α [1].

1.3 Global dynamic optimization

Deterministic global *dynamic* optimization methods have been successfully developed for about a decade [48, 114, 116, 117, 162–164, 171, 172, 174, 177, 186–188]. Although these methods are valid and guarantee finding a global optimum, at the outset of this thesis they were limited to problems with up to about 5 parameters and 5 state variables. In Chapter 2, we present computational results testing the methods from [172, 174, 177].

Broadly, deterministic global dynamic optimization methods fall into three categories. One method is known as discretize-then-bound, or Taylor models, in which the Taylor expansion in time of the solution of the ODE is constructed then interval techniques are used to bound the Taylor model. This class dates back to Moore's thesis in 1962 [134] and many enhancements have been made since then [86, 114, 116, 117, 135, 163, 164]. A second approach is a dynamic extension of αBB [148]. Applying this approach rigorously requires bounding the second-order sensitivities of the dynamic system, so that a large number of equations must be numerically integrated and bounded, with commensurate computational cost and overestimation. A third approach, which we refer to as the auxiliary

ODE approach, relies on auxiliary ODE systems constructed such that their solutions give bounds [171, 172, 186], affine relaxations [187], or nonlinear relaxations [174, 177] of the solution of the parametric ODE. We focus on the auxiliary ODE approach.

1.4 Outline of the thesis

Chapters 2 and 3 focus on global dynamic optimization, which combines the ideas from Sections 1.1 and 1.2 to certify global optimality for dynamic optimization problems. Chapter 4 focuses on the use of time-varying (dynamic) experiments in microreactors to discriminate between candidate models and identify the best-fit parameters of the models.

Chapter 2 describes software created for deterministic global dynamic optimization based on auxiliary ODEs. We give an overview of the implementation details and numerical results, comparing to previous work. A primary goal of this thesis was to reduce the CPU requirements for software for global dynamic optimization thereby making it suitable for larger problems. Faster CPU times were achieved for many benchmark problems by implementing new methods [172, 174, 177] and improving heuristics. We have solved practical problems with up to 9 parameters and 7 state variables and a test problem inspired by PDE discretization with up to 41 state variables.

Chapter 3 is the most significant theoretical contribution of this thesis. There, the convergence order and prefactor are analyzed for two convex relaxation methods [174, 177] used in global dynamic optimization. It is shown that although both methods analyzed guarantee second-order convergence, the newer method [174] dominates the older [177]. The newer method always gives a smaller convergence prefactor than the older method, sometimes vastly smaller. For the relaxations from the older method, once a certain level of conservatism has been reached, that conservatism can never decrease. However, using the newer relaxation method, the relaxations can actually shed conservatism as the independent variable in the ODE (usually time) increases. When coupled with the fact that the state bounding method gives first-order convergence this analysis gives rise to a critical parameter interval diameter w_{crit}. For parameter intervals smaller than w_{crit}, the empirical convergence behavior is second-order, whereas it is first order for parameter intervals larger than w_{crit}.

For the older relaxation method, w_{crit} tends to decrease rapidly with increasing time, making it more and more difficult to achieve second-order empirical convergence. For the newer relaxation method, w_{crit} tends to be a much weaker function of time, making the highly-desired second-order empirical convergence much more probable.

Chapter 4 describes the design and execution of time-varying experiments to estimate parameters for a chemical reaction in microreactors. Using time-varying experiments in microreactors rather than the traditional microreactor experimentation approach of setting conditions and waiting for steady state before recording a data point has the potential to significantly decrease the time and material required to accurately discriminate between candidate models and estimate model parameters in microreactors. The difficulty with this idea that has prevented its adoption for microreactor experiments is the requirement to simulate the solution of a PDE within an optimization routine. When using an appropriate spatial discretization and a suitable ODE simulator, however, a half-day-long experiment can be simulated in a matter of minutes on a modern personal computer, making dynamic experimental design entirely feasible, even with the embedded approximate PDE solution. Whenever running experiments, National Instruments LabView was used to automatically perform the time-varying experiment, setting flow rates and the reactor temperature over time and recording data from a Fourier Transform Infrared (FTIR) spectroscopic flow cell.

Chapter 5 gives a few overarching conclusions from the thesis and outlook for future research in the area. Appendix A gives a brief overview of the convergence of the McCormick relaxation of the product operation when only one of the two variables is partitioned. Appendix B is an article that we published regarding the economics of continuous versus batch production of a large-production-volume small-molecule pharmaceutical tablet.

Chapter 2

dGDOpt: Software for deterministic global optimization of nonlinear dynamic systems

Abstract

Dynamic systems are ubiquitous in nature and engineering. Two examples are the chemical composition in a living organism and the time-varying state of a vehicle. Optimization of dynamic systems is frequently used to estimate model parameters or determine control profiles that will achieve the best possible result, such as highest quality, lowest cost, or fastest time to reach some endpoint. Here we focus on dynamic systems that can be modeled by nonlinear ordinary differential equations (ODEs). We implemented methods based on auxiliary ODE systems that are theoretically guaranteed to find the best possible solution to an ODE-constrained dynamic optimization problem within user-defined tolerances. Our software package, named dGDOpt, for deterministic Global Dynamic Optimizer, is available free of charge from the authors. The methods have been tested on problems with up to nine parameters and up to 41 state variables. After adjusting for the differences in CPU performance, the methods implemented here give up to 50 times faster CPU times than the Taylor model-based methods implemented in Sahlodin (2013) for one parameter estimation problem and up to twice as fast for one optimal control problem. Again adjusting for differences in CPU performance, we achieved CPU times similar to or better than Lin and Stadtherr (2006) on two chemical kinetic parameter estimation problems and better scaling of CPU time with the number of control parameters on an optimal control problem than Sahlodin (2013) and Lin and Stadtherr (2007).

Keywords: *dynamic optimization, differential inequalities, global optimization, McCormick relaxations, nonconvex optimization, optimal control, state bounds*

For background, see Chapter 1.

2.1 Methods

All methods used the branch-and-bound framework. We implemented domain reduction techniques, so the method could more aptly be called branch-and-reduce [160], but we use the more widely-recognized term branch-and-bound.

2.1.1 Bounds and relaxations

We require convex relaxations of the objective and constraint functions to solve the lower-bounding problems in the branch-and-bound framework. To obtain such relaxations using the McCormick relaxation technique [125], we require time-varying lower and upper bounds as well as convex and concave relaxations of the solutions of the ODEs.

Three methods were used to bound the solutions of ordinary differential equations: natural bounds [186] and two convex polyhedral bounding methods, given by [172, Equation (6)] and [172, Equation (7)]. We will refer to the bounding methods as NatBds, ConvPoly1, and ConvPoly2, respectively. All three bounding methods rely on some amount of *a priori* information on the solutions of the ODEs. NatBds prune the state space using an interval X^N that is known to contain the solution of the ODE for all time. These state bounds can come from conservation relations, such as the fact that the total mass of any component of a closed system can never exceed the total mass of the system and concentrations, masses, pressures, and absolute temperatures must always be nonnegative. If the model is physically correct, these will also be mathematical properties of the model that can be verified, for example through viability theory [7]. ConvPoly1 and ConvPoly2 use the set X^N in addition to known affine invariants or affine bounds in state space defining a convex polyhedron G known to contain the solution of the ODE for all time. In the case that an interval X^N is used in place of the convex polyhedral set G, the ConvPoly methods reduce to NatBds. In all cases, ConvPoly2 is guaranteed to be at least as tight as, and possibly tighter than, ConvPoly1, which in turn is guaranteed to be at least as tight as NatBds. However, ConvPoly1 is computationally cheaper than ConvPoly2 by a factor of $2n_x$ and

NatBds are computationally cheaper than ConvPoly1 when G is not an interval. Because of this tradeoff, the optimal method between NatBds, ConvPoly1, and ConvPoly2 is problem-dependent. NatBds and the pruning portion of the ConvPoly methods can also be used to tighten the bounds after integration but before calculation of the convex relaxation of the objective function. This post-integration bounds-tightening step has very little cost compared to performing the bounding during numerical integration and can significantly tighten bounds, so it is always used.

Convex relaxations of the states for the ODEs were computed using the affine relaxation (AR) method of Singer and Barton [187] and the two nonlinear methods of Scott and Barton [174, 177]: the earlier relaxation-amplifying dynamics (RAD, [177]) and the later relaxation-preserving dynamics (RPD, [174]). For any given test problem, a single template function for the ODE vector field was used, so that it could be evaluated using both real arithmetic and McCormick arithmetic. Using a template function in this way eliminates a potential source of errors: there is no need to create separate vector field functions for the lower-bounding and upper-bounding problems in each example. After integration, the nonlinear relaxations to the states were linearized using sensitivity analysis. By linearizing, integration is only necessary for the first lower-bounding function evaluation per lower-bounding subproblem; subsequent function evaluations in the lower-bounding problem were computed using the linearized values for the state relaxations. Since the objective functions can depend nonlinearly on the states, the final objective function can still be nonlinear, even when the states are linearized. For example, note that least-squares parameter estimation problems depend nonlinearly on the state variables. The relaxations were always linearized at the midpoint of the parameter bounds, $\mathbf{p}^{mid}$ (see Table 2.1). Since the relaxations are convex on the (compact) decision space, a supporting hyperplane must exist at any $\mathbf{p}$ in the interval over which the relaxations are constructed. Once we generate a supporting hyperplane using a subgradient to the relaxation, any state value on that hyperplane gives an affine underestimator for on the relaxation and hence a bound on the solution to the original ODE. A state value on the hyperplane can be computed using the value of the convex relaxation of the solution to the original ODE and the subgradient of the convex relaxation. In almost all problems, we found that linearizing the relaxations gave faster CPU times for solving the global optimization problems than using the nonlinear relaxations directly. That is, the additional cost of numerical integration for every function evaluation in each lower-bounding problem was not overcome by a sufficient decrease in the number of B&B nodes.

Singer and Barton's theory [187] has two significant drawbacks: (i) it requires selecting a reference trajectory, which strongly affects the strength of the relaxations and (ii) it required event detection and discontinuity locking for certain reference trajectories and problems. Drawback (i) implies that while a given global optimization problem may be solved efficiently with certain reference trajectories, the best reference trajectory is not known *a priori* and it may be necessary to try multiple reference trajectories to solve the problem in a reasonable CPU time. Here, our implementation eliminates drawback (ii) of Singer and Barton's method [187]. To explain further: in [187], a discontinuity-locking scheme was used. The convex and concave relaxations of the right-hand side may, in general, be nonsmooth since they can contain min and max functions of two variables and mid functions which return the middle value of three scalars. At certain times during integration, the argument selected by the min, max, or mid may be arbitrary since they are equal. When such a case is implemented on a finite-precision computer, the argument selected may switch back and forth many times in the numerical integration due to round-off error. Since either argument is valid, the relaxations remain valid, but the integrator may switch between modes arbitrarily often, causing a very large number of integration steps. Singer addressed this by detecting integration "chattering", when the min, max, or mid alternated in quick succession, and locking the switch into an arbitrary mode by adding a small positive constant to one of the arguments. We have found that chattering only occurs for particular state reference trajectories. Since any reference trajectory in the current parameter bounds and time-varying state enclosure is valid as long as it does not depend on the current value of the parameter, we perturbed the reference trajectory for the state slightly and avoided chattering completely. The exact reference trajectories we used are given in Table 2.1. Note that we perturbed the state reference values but not the parameter reference values. These slight perturbations eliminated the need for a discontinuity locking, making the implementation simpler and the resulting CPU times faster.

Table 2.1: Meaning of reference trajectories

Abbreviation	Value used
$\mathbf{p}^{mid}$	$0.5(\mathbf{p}^L + \mathbf{p}^U)$
$\mathbf{x}^{L*}$	$(1.0 - 10^{-8})\mathbf{x}^L + 10^{-8}\mathbf{x}^U$
$\mathbf{x}^{mid*}$	$(0.5 + 10^{-8})\mathbf{x}^L + (0.5 - 10^{-8})\mathbf{x}^U$
$\mathbf{x}^{U*}$	$10^{-8}\mathbf{x}^L + (1.0 - 10^{-8})\mathbf{x}^U$

The origin of chattering can be understood in the following way. All of the problems tested by Singer and Barton [185, 187] have products $(x, y) \mapsto xy$ in the factored representations of their vector fields. The McCormick rule for the convex (resp. concave) relaxation of a product is nonsmooth on the line connecting $x^L y^U$ to $x^U y^L$ (resp. $x^L y^L$ to $x^U y^U$). Therefore, both the convex and concave relaxations are nonsmooth at $(\frac{x^L+x^U}{2}, \frac{y^L+y^U}{2})$, so that the subgradient is discontinuous at that point. Therefore, the computed value of the subgradient can switch back and forth depending on roundoff error in a finite-precision computer. This subgradient is used to compute the vector field for the ODE that generates state relaxations, therefore a discontinuous subgradient at $(\frac{x^L+x^U}{2}, \frac{y^L+y^U}{2})$, coupled with roundoff error to slightly perturb the arguments of the vector field, can yield a discontinuous vector field when implemented on a finite-precision computer, which can in turn yield chattering in numerical integration observed in [185, 187] when using the exact midpoint for the reference trajectory. Perturbing the reference trajectory sufficiently far away from points of nonsmoothness on the relaxations (cf. Table 2.1) yields a continuous ODE vector field for generating state relaxations even with numerical error and fixes the chattering problem in all cases that we studied. Whereas in the affine relaxation theory [187], the reference trajectory is fixed in time relative to the state bounds (Table 2.1), which causes the chattering problem. On the other hand, for the nonlinear relaxation theories (RAD and RPD), there is no reference trajectory, so the point in state space at which the McCormick relaxation for the vector field is evaluated varies with time relative to the state bounds. This makes chattering occur much less frequently for RAD and RPD.

A subgradient of the objective function is computed in the following way. Sensitivity analysis with the staggered corrector method [69] is used during integration to compute the subgradients of the convex and concave relaxations to the states, which are then propagated to a subgradient of the objective function using the operator overloading library MC++ [47]. MC++ is the successor to libMC, which is described in detail in [130]. Error control for the sensitivities is enabled in the integrator, which increases the cost of integration but guarantees accuracy of sensitivity information, which is needed for accurate subgradient information, linearizations of the objective, and domain reduction (§2.1.2). In most cases, the vector fields for the sensitivity systems were computed using the algorithmic differentiation (AD) library FADBAD++ [22] and the subgradient capability of MC++. In FADBAD++, we used the stack-based allocation by pre-specifying the number of parameters to which derivatives are taken. This is significantly faster than the dynamically-allocated alternative. Otherwise, AD objects would be created and their derivatives dynamically allocated in each evaluation of the right-hand side function. If, during the course of integration, the state relaxations leave the state bounds, the sensitivity of the offending relaxation is reset to zero (see Proposition 2.1.1).

Next we argue the validity of using subgradients of the vector fields of the relaxation systems to generate subgradients of the relaxations of the solutions of the ODEs. RAD [177] satisfy the hypotheses of [52, Theorem 2.7.3], so that integrating a subgradient of the vector fields for the convex and concave relaxations yields a subgradient of the relaxations of the solution of the ODE. In the affine relaxation (AR) theory [187], the subgradient of the McCormick relaxation is always evaluated at a reference trajectory within the state and parameter bounds. Therefore, as long as the reference trajectory at which the vector field for the relaxation system is evaluated does not stay on a point of nonsmoothness for finite time, [221, Theorem 3.2.3] guarantees that the resulting sensitivity information will give a partial derivative (and therefore also a subgradient) of the relaxations of the solution of the ODE at each point in time. See also [52, Theorem 7.4.1]. By perturbing the reference trajectory away from points of nonsmoothness in the vector field of the ODE used to generate the state relaxations, we obtained a system for which the ODE vector field does not stay on a point of nonsmoothness for finite time (a so-called *sliding mode*) and [221, Theorem 3.2.3] guarantees that we obtain a subgradient of the state relaxations. For RPD, it is again valid to use the subgradient of the vector field to calculate a subgradient of the solution of the ODE relaxation system, provided there is no sliding mode, because

[221, Theorem 3.2.3] again tells us that we obtain a partial derivative which is guaranteed to be a subgradient. We have noticed that when sliding modes occur, numerical integration tends to take an excessive number of steps, therefore failing, and returning $-\infty$ for a lower bound, so that the node would be partitioned and revisited, and this process repeated until there was no sliding mode. However, to make the solver's implementation of RPD rigorous, we need to be able to rigorously guarantee that there are no sliding modes so that the subgradient information is accurate. One way to do this would be to use the necessary conditions for a sliding mode to arise as derived by [95]. The key steps are as follows: (i) reconsider the McCormick relaxation as an *abs-factorable function* [77], in which all nonsmoothness in the factored representation arises due to absolute value functions, (ii) generate a vector containing the values of the arguments of all absolute value functions in the abs-factorable representation, (iii) employ event detection (rootfinding) to determine when each of the arguments of the absolute value functions crosses zero, and (iv) at each zero-crossing, determine whether there is a second, distinct, root function that is also within some numerical tolerance of zero and has a derivative within some tolerance of zero. If the situation in (iv) never arises, then there is no sliding mode and the subgradient method furnishes a partial derivative. If the situation in (iv) does arise, then there could be a sliding mode and we cannot guarantee that the subgradient information for the relaxations of the solution of the ODE is valid. In that case, we could complete the numerical integration for the current node and use the interval bounds (only) to compute a lower bound, setting the vector field for the relaxations equal to zero so that any potential sliding modes do not slow numerical integration.

To generate an abs-factorable representation of a McCormick relaxation for (i) above, observe that the nonsmoothness in McCormick relaxations arises due to min, max, and mid functions. The computations for min and max can be reformulated using the absolute value function with the following well-known identities:

$$\begin{aligned}\min\{x,y\} &= \frac{1}{2}(x+y) - \frac{1}{2}|x-y|,\\ \max\{x,y\} &= \frac{1}{2}(x+y) + \frac{1}{2}|x-y|.\end{aligned}$$

The mid function can be reformulated in terms of min and max:

$$\text{mid}\{x, y, z\} = \max\left\{\min\left\{x, y\right\}, \max\left\{\min\left\{y, z\right\}, \min\left\{x, z\right\}\right\}\right\},$$

so that, when the absolute value forms are employed for min and max, we can obtain an abs-factorable representation for any McCormick relaxation. The only remaining difficulty is to modify MC++ so that it also outputs the values of the arguments of each absolute value function in the factored representation for (ii) above.

For more information about the McCormick-based [125] ODE bounding and relaxation theory, see [170, 174, 178].

2.1.2 Domain reduction

Domain reduction, also known as *range reduction* [129, 159, 160, 195], techniques can be used to eliminate subsets of the search space from consideration. See [195] for a framework that unifies many of the domain-reduction methods, including those used here. In some cases, domain reduction greatly accelerates convergence. Two types of domain reduction tests have been used: Tests 1 & 2 from [159] can be used only when the solution of a lower-bounding subproblem lies on a parameter bound; *probing* [159, Tests 3 & 4] is more computationally intensive but can be applied for any node in the branch-and-bound tree, by solving up to $2n_p$ different optimization problems at any node. In dGDOpt, we use affine relaxations to the states based on a single numerical integration of the auxiliary ODE system. These affine relaxations to the states are used in both the lower-bounding problem, and the $2n_p$ probing problems, greatly reducing the cost of probing by performing a single integration instead of $2n_p + 1$.

All four range-reduction tests exploit the following idea: if there is a region of the search space for which the convex underestimator for the objective function has a value greater than or equal to the best known upper bound for the problem, that region of the search space can be eliminated, for it cannot contain a better solution than the feasible solution we have already found.

In the test problems in the literature, enabling probing can reduce or increase CPU time required to solve a given problem. In the four examples in [159, Table 5], probing *increased* the CPU time by a factor of 1.2–1.5. In the 27 examples in [160, Table V], the ratio of CPU time *with* probing to that *without* ranged from 0.4 to 5.7, with a median value of 1.07. In all cases, the number of nodes decreased or remained the same when probing was enabled. Sometimes a very large reduction in the number of nodes is possible by enabling probing: in one problem listed in [160, Table V], the number of nodes required was reduced from 49 to 3.

With all of the possible combinations of bounding, relaxation, domain reduction, other heuristics, and problem structures it is nearly impossible to say with certainty which combination is best overall. For this reason, we focus on one "base case" of method choices that we consider to be good options for typical problems and occasionally explore the effect of using different choices for particular problems. Focusing on one base case also ensures a fair comparison by changing only one aspect of the method at a time.

2.1.3 Implementation details

The choice of reference trajectory $(\mathbf{x}^{ref}, \mathbf{p}^{ref})$ for the Singer relaxation method can have a very large impact on the performance of the method, yet it is impossible to know in advance which reference trajectory will be best. To account for this drawback of the method, we chose to use the midpoint reference trajectory throughout these case studies for linearizing both the Singer and Scott relaxations. A practitioner wanting to solve a global dynamic optimization problem only wants to solve it once rather than trying several different reference trajectories. Thus, we think that running the test suite with a single reference trajectory better emulates the performance likely to be encountered in practice. In our preliminary tests, we found that the midpoint tends to be either the best reference trajectory, or not much worse than the best. In contrast, other reference trajectories such as $(\mathbf{x}^{L*}, \mathbf{p}^{L*})$ or $(\mathbf{x}^{U*}, \mathbf{p}^{U*})$ can be much slower than the midpoint reference trajectory $(\mathbf{x}^{mid*}, \mathbf{p}^{mid*})$. For example, see Table 2.10.

2.1.3.1 Nonlinear local optimizer

The sequential quadratic programming (SQP) optimization solver SNOPT [74] version 7.2 was always used as the local optimizer for the upper-bounding problem and used except where otherwise noted for the lower-bounding problem. This sequential quadratic programming (SQP) code has been preferred for optimal control problems because it uses relatively few objective function, constraint, and gradient evaluations. Evaluations of all three of those quantities can be very expensive for dynamic optimization because they depend on the numerical solution of an ODE. The method only requires first derivatives, which it uses in a BFGS-type update of the approximate Hessian. First derivatives of the solutions of an ODE with respect to parameters can be calculated relatively efficiently and automatically [69, 123], whereas second derivatives are more expensive and automated implementations are less widely available.

2.1.3.2 Linear local optimizer

The linear programming (LP) solver CPLEX 12.4 was used to minimize the lower-bounding objective for a few test cases. The objective function was the supporting hyperplane for the nonlinear convex relaxation generated using the function value and subgradient at $\mathbf{p}^{mid}$. The interval lower bound for the objective function from the state bounds and interval arithmetic was used as a lower-bounding constraint for the objective function.

2.1.3.3 Event detection scheme for relaxation-preserving dynamics

Integrating the ODEs used to generate relaxations by relaxation-preserving dynamics requires that the state relaxations never leave the state bounds. To ensure this, we must identify the exact event time t_e when $x_i^{cv}(t_e, \mathbf{p}) = x_i^L(t_e)$ or $x_i^{cc}(t_e, \mathbf{p}) = x_i^U(t_e)$ for each applicable i. We do this using the built-in rootfinding features of CVODES, with the event detection scheme in [170, §7.6.3]. First the initial condition is checked to set the proper mode for the binary variables, then the integration is run with rootfinding enabled using the root functions and state vector fields given in [170, §7.6.3]. An analogous scheme is also used to detect when the time-varying state bounds leave the time-invariant natural state bounds (if any).

Whenever one of the relaxations reaches the bounds, there is typically a jump in the sensitivities $\frac{\partial \mathbf{x}^{cv/cc}}{\partial \mathbf{p}}$ for which we must account [71]. When the integration event is detected, integration halts, the sensitivity is reset to its new value, and the sensitivity calculation is reinitialized from that point.

Proposition 2.1.1. *For relaxation-preserving dynamics, for some $i = 1, \ldots, n_x$, $\frac{\partial x_i^{cv}}{\partial \mathbf{p}}$ jumps to $\mathbf{0}$ when x_i^{cv} reaches the bound x_i^L from above. Similarly, when x_i^{cc} reaches x_i^U from below, $\frac{\partial x_i^{cc}}{\partial \mathbf{p}}$ jumps to $\mathbf{0}$.*

Proof. Consider the case when $x_i^{cv}(t, \mathbf{p})$ approaches $x_i^L(t)$ from above for some i. Then $f_i^{cv,(j)} = \frac{\partial x_i^{cv,(j)}}{\partial t}$ and $f_i^{cv,(j+1)} = \frac{\partial x_i^{L,(j)}}{\partial t}$, where j denotes the current epoch of the dynamic system, and j is incremented by 1 each time there is an event. In our problems, the only events occur when $x_i^{cv}(\widehat{t}, \mathbf{p}) = x_i^L(\widehat{t})$ or $x_i^{cc}(\widehat{t}, \mathbf{p}) = x_i^U(\widehat{t})$ for some i and some $\widehat{t}$. In the rest of this proof, except where we explicitly declare a function, we refer to functions evaluated at points. We have omitted the arguments for readability but the functions are understood to be evaluated at the points shown below.

$\frac{\partial g_{j+1}^{(j)}}{\partial \dot{\mathbf{x}}^{(j)}}$, $\frac{\partial g_{j+1}^{(j)}}{\partial \mathbf{x}^{(j)}}$, $\frac{\partial g_{j+1}^{(j)}}{\partial p_k}$, and $\frac{\partial g_{j+1}^{(j)}}{\partial t}$ are evaluated at $(\dot{\mathbf{x}}^{(j)}(t_f^{(j)}, \mathbf{p}), \mathbf{x}^{(j)}(t_f^{(j)}, \mathbf{p}), \mathbf{p}, t_f^{(j)})$,
$\frac{\partial \mathbf{f}^{(j)}}{\partial p_k}$ and $\frac{\partial \mathbf{f}^{(j)}}{\partial t}$ are evaluated at $(t_f^{(j)}, \mathbf{x}^{(j)}(t_f^{(j)}, \mathbf{p}), \mathbf{p})$,
$\frac{\partial \mathbf{x}^{(j)}}{\partial p_k}$, $\frac{\partial \mathbf{x}^{(j)}}{\partial t}$, and $\frac{\partial \mathbf{x}_i^{cv,(j)}}{p_k}$ are evaluated at $(t_f^{(j)}, \mathbf{p})$,
$\frac{\partial \mathbf{x}_i^{L,(j)}}{p_k}$ is evaluated at $(t_f^{(j)})$.

The following relation gives the jumps in sensitivities for an ODE with continuous states [71, Eq. (57)]:

$$\frac{\partial x_i^{cv,(j+1)}}{\partial p_k} - \frac{\partial x_i^{cv,(j)}}{\partial p_k} = -(f_i^{cv,(j+1)} - f_i^{cv,(j)})\frac{\mathrm{d}t}{\mathrm{d}p_k}. \tag{2.1}$$

$\frac{\mathrm{d}t}{\mathrm{d}p_k}$ is given by [71, Eq. 50]:

$$\frac{\mathrm{d}t}{\mathrm{d}p_k} = -\frac{\frac{\partial g_{j+1}^{(j)}}{\partial \dot{\mathbf{x}}^{(j)}}\frac{\partial \mathbf{f}^{(j)}}{\partial p_k} + \frac{\partial g_{j+1}^{(j)}}{\partial \mathbf{x}^{(j)}}\frac{\partial \mathbf{x}^{(j)}}{\partial p_k} + \frac{\partial g_{j+1}^{(j)}}{\partial p_k}}{\frac{\partial g_{j+1}^{(j)}}{\partial \dot{\mathbf{x}}^{(j)}}\frac{\partial \mathbf{f}^{(j)}}{\partial t} + \frac{\partial g_{j+1}^{(j)}}{\partial \mathbf{x}^{(j)}}\frac{\partial \mathbf{x}^{(j)}}{\partial t} + \frac{\partial g_{j+1}^{(j)}}{\partial t}}, \quad \forall k \in \{1, \dots, n_p\}, \tag{2.2}$$

The discontinuity function

$$g_{j+1}^{(j)} : (\dot{\mathbf{z}}^{(j)}, \mathbf{z}^{(j)}, \mathbf{p}, t) \mapsto z_i^{cv(j)} - z_i^{L(j)} - b_i^{cv}\epsilon,$$

where $\mathbf{z}$ is a dummy variable to avoid confusion with $\mathbf{x}$, implies that

$$\frac{\partial g_{j+1}^{(j)}}{\partial \dot{\mathbf{x}}^{(j)}} \equiv \mathbf{0}, \quad \frac{\partial g_{j+1}^{(j)}}{\partial \mathbf{p}} \equiv \mathbf{0}, \quad \frac{\partial g_{j+1}^{(j)}}{\partial t} \equiv 0.$$

Therefore, (2.2) reduces to

$$\frac{\mathrm{d}t}{\mathrm{d}p_k} = -\frac{\frac{\partial g_{j+1}^{(j)}}{\partial \mathbf{x}^{(j)}}\frac{\partial \mathbf{x}^{(j)}}{\partial p_k}}{\frac{\partial g_{j+1}^{(j)}}{\partial \mathbf{x}^{(j)}}\frac{\partial \mathbf{x}^{(j)}}{\partial t}} = -\frac{\frac{\partial x_i^{cv,(j)}}{\partial p_k} - \frac{\partial x_i^{L,(j)}}{\partial p_k}}{\frac{\partial x_i^{cv,(j)}}{\partial t} - \frac{\partial x_i^{L,(j)}}{\partial t}}, \quad \forall k.$$

Substituting this information into (2.1), we obtain:

$$\begin{aligned}
\frac{\partial x_i^{cv,(j+1)}}{\partial p_k} - \frac{\partial x_i^{cv,(j)}}{\partial p_k} &= -\left(\frac{\partial x_i^{L,(j)}}{\partial t} - \frac{\partial x_i^{cv,(j)}}{\partial t}\right)\left(-\frac{\frac{\partial x_i^{cv,(j)}}{\partial p_k} - \frac{\partial x_i^{L,(j)}}{\partial p_k}}{\frac{\partial x_i^{cv,(j)}}{\partial t} - \frac{\partial x_i^{L,(j)}}{\partial t}}\right), \quad \forall k \\
\implies \frac{\partial x_i^{cv,(j+1)}}{\partial \mathbf{p}} - \frac{\partial x_i^{cv,(j)}}{\partial \mathbf{p}} &= -\left(\frac{\partial x_i^{cv,(j)}}{\partial \mathbf{p}} - \frac{\partial x_i^{L,(j)}}{\partial \mathbf{p}}\right), \\
&= -\frac{\partial x_i^{cv,(j)}}{\partial \mathbf{p}},
\end{aligned}$$

where $\frac{\partial x_i^{L,(j)}}{\partial \mathbf{p}} = \mathbf{0}$ always. Therefore,

$$\frac{\partial x_i^{cv,(j+1)}}{\partial \mathbf{p}} = \frac{\partial x_i^{cv,(j)}}{\partial \mathbf{p}} - \frac{\partial x_i^{cv,(j)}}{\partial \mathbf{p}} = \mathbf{0}.$$

The proof is analogous when x_i^{cc} reaches x_i^U from below. □

Proposition 2.1.2. *The sensitivity* $\frac{\partial x_i^{cv}}{\partial \mathbf{p}}$ *does not have a jump at the point where* x_i^{cv}

becomes greater than x_i^L*, and similarly for* x_i^{cc} *and* x_i^U*.*

Proof. This can be seen by a proof similar to that of Proposition 2.1.1. □

The relaxed objective function generated using McCormick relaxations, while convex, may be nonsmooth. In practice, we have found this is particularly problematic for the local solver SNOPT when working with the nonlinear relaxations generated by RPD. SNOPT is designed to solve smooth problems, so this is not unexpected. This problem was averted by working with the affine relaxations to the states. The affine state relaxations are still nonsmooth if the state bounds are used to cut them, so if SNOPT gave too many return codes indicating numerical difficulties, the state bounds were no longer used to cut the relaxations for the rest of the global optimization, which mitigated the problem. The problem could be made smooth by introducing additional variables and constraints to the local optimization problem in SNOPT. In particular, the additional variables in the local optimizer would be:

$$\begin{aligned}
&z_{i,j}^{cv}, \quad \forall (i,j) \in \{1,\dots,n_x\} \times \{1,\dots,n_{times}\},\\
&z_{i,j}^{cc}, \quad \forall (i,j) \in \{1,\dots,n_x\} \times \{1,\dots,n_{times}\},
\end{aligned}$$

as well as auxiliary variables for the smooth reformulation of the McCormick relaxation of the objective function. The additional constraints in the local optimizer would be:

$$\begin{aligned}
&z_{i,j}^{cv} \geq x_i^L(t_j), \quad \forall (i,j) \in \{1,\dots,n_x\} \times \{1,\dots,n_{times}\},\\
&z_{i,j}^{cc} \leq x_i^U(t_j), \quad \forall (i,j) \in \{1,\dots,n_x\} \times \{1,\dots,n_{times}\},\\
&z_{i,j}^{cv} \geq x_i^{cv}(t_j,\widehat{\mathbf{p}}) + \left(\frac{\partial x_i^{cv}}{\partial \mathbf{p}}(t_j,\widehat{\mathbf{p}})\right)^{\mathrm{T}} (\mathbf{p}-\widehat{\mathbf{p}}), \quad \forall (i,j) \in \{1,\dots,n_x\} \times \{1,\dots,n_{times}\},\\
&z_{i,j}^{cc} \leq x_i^{cc}(t_j,\widehat{\mathbf{p}}) + \left(\frac{\partial x_i^{cc}}{\partial \mathbf{p}}(t_j,\widehat{\mathbf{p}})\right)^{\mathrm{T}} (\mathbf{p}-\widehat{\mathbf{p}}), \quad \forall (i,j) \in \{1,\dots,n_x\} \times \{1,\dots,n_{times}\},
\end{aligned}$$

as well as additional equations to represent the smooth reformulation of the McCormick relaxation of the objective function and constraints in terms of the $z_{i,j}^{cv/cc}$ and $x_i^{L/U}(t_j)$.

For some problems, we also ran the test problems using the linear programming solver CPLEX and saw improved CPU times.

2.1.3.4 Preprocessing

To increase the chances of a strong upper-bound, the upper-bounding problem is solved 30 times using random initial guesses before any lower-bounding problems are solved. With a strong upper-bound available at the root node for the lower-bounding problem, domain reduction can be much more effective.

2.1.3.5 Branch-and-bound

A branch-and-bound library internal to our research group was used. Tolerances specified in the local optimizer for each upper- and lower-bounding problem were always at least 100 times tighter than those in the branch-and-bound framework; tolerances in the integrator were always set at least 100 times tighter than those in the local optimizer. Unless otherwise noted, the variable with the largest absolute diameter ($\max_i p_i^U - p_i^L$) was chosen for branching. We refer to this as AbsDiamBV. We also tested a different heuristic for selecting the variable to branch on, which we refer to as GradBV. Let $\sigma_{h^{cv},p_i}(\mathbf{p}^{mid})$ be a subgradient of the convex relaxation to the objective function with respect to decision variable p_i at the midpoint of the current node. For GradBV we branch on variable $i \in \arg\ \max_i \left|\sigma_{h^{cv},p_i}(\mathbf{p}^{mid})\right| (p_i^U - p_i^L)$ except when level of B&B tree is evenly divisible by 3, in which case the variable with the largest absolute diameter is chosen for branching. We did not find a description of GradBV in the global optimization literature, but it is related to the idea mentioned in [160, §4.1.2]: "Select a variable p_j which is 'mostly responsible' for the difference [between upper and lower bounds on the current node]". In our case, it selects the variable that contributes the most to the variation of the affine underestimator on the current node, rather than the difference between the upper and lower bounds. The GradBV idea also appears to be similar to the ideas for branching variable selection developed in [181, §3.5] and [194, §6.2.1].

We always choose the node with the least lower bound to process next and always branch at the midpoint of the chosen branching variable (bisection).

2.1.3.6 Hardware, operating system, and compiler

The algorithm was implemented in C++ on Ubuntu Linux using GCC as the compiler with the -O2 optimization flag and allocated a single core of an Intel Xeon W3550 3.07 GHz CPU and 1.0 GB of RAM. There was no parallelization scheme—the code was implemented as a single thread.

2.2 Numerical results

In this section, subproblem counts and CPU times for the different bounding and relaxation methods are given for several test problems from the literature. All parameter estimation problems are formulated as a minimization of the unweighted sum of squared differences between experimental data and simulation:

$$\min_{\mathbf{p}} \sum_i \sum_j (x_{i,j}^{\text{meas}} - x_j(t_i, \mathbf{p}))^2,$$

where i indexes times at which data were measured, j indexes state variables for which experimental data is available at the current time point, and $\mathbf{x} : [t_0, t_f] \times P \to \mathbb{R}^{n_x}$ is given by the solution of the ODE:

$$\begin{aligned}
&\dot{\mathbf{x}}(t, \mathbf{p}) = \mathbf{f}(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}), \quad \forall t \in (t_0, t_f],\\
&\mathbf{x}(t_0, \mathbf{p}) = \mathbf{x}_0(\mathbf{p}).
\end{aligned}$$

Throughout this chapter, we use the abbreviations in Table 2.2. For post-integration pruning, the most advanced method possible was always used (NatBds or ConvPoly1), since it adds very little to the cost per node and always gives equal or tighter bounds, so always produces similar or faster CPU times in the overall global optimization procedure. Note that while ConvPoly2 gives the tightest bounds when used during integration, the flattening step of ConvPoly2 is not valid for post-integration pruning, making ConvPoly1 the tightest possible bounding method post-integration.

Table 2.2: List of abbreviations

AbsDiamBV	Branch on the variable with the largest absolute diameter ($i \in \max_i p_i^U - p_i^L$).
AR	Affine relaxation theory for nonlinear ODEs [185, 187].
ConvPoly1	Prune-first convex polyhedral bounding technique [172, Equation (6)].
ConvPoly2	Flatten-first convex polyhedral bounding technique [172, Equation (7)].
NaïveBds	Similar to differential inequalities, but no flattening step.
GradBV	A rule for selecting the variable to branch on. See §2.1.3.5.
IBTM	Interval bounds from Taylor models [116, 162].
LBP	Lower-bounding problem.
NatBds	Natural bounds [186].
PRMCTM	Polyhedral relaxations of McCormick-Taylor models [162].
PRTM	Polyhedral relaxations of Taylor models [162].
RAD	Relaxation-amplifying dynamics [170, 177].
RPD	Relaxation-preserving dynamics [170, 174].
UBP	Upper-bounding problem.

2.2.1 Reversible series reaction parameter estimation

The first problem is a four-parameter estimation problem for the first-order reversible chain reaction A $\leftrightharpoons$ B $\leftrightharpoons$ C from [201], also solved by [67, 162, 187]. The problem has been solved with four different sets of data: (i) noise-free data for all three states from [67], (ii) data for states x_1 and x_2 with noise added from [67], (iii) data for all three states with noise added from [67], and (iv) the data from [162], which differs from (i)–(iii) above.

2.2.1.1 Using noise-free data

Noise-free data generated using parameter values of $(4.0, 2.0, 40.0, 20.0)$ were taken from [67]. Note that the equation for $\dot{x}_3$ in [67] has a typographical error, but their numerical results are consistent with the statement below. The ODE model is:

$$\begin{aligned}
&\dot{x}_1 = -p_1 x_1 + p_2 x_2,\\
&\dot{x}_2 = p_1 x_1 - (p_2 + p_3) x_2 + p_4 x_3,\\
&\dot{x}_3 = p_3 x_2 - p_4 x_3,\\
&\mathbf{x}(0) = (1, 0, 0),\\
&\mathbf{p} \in [0, 10]^2 \times [10, 50]^2,\\
&t \in [t_0, t_f] = [0, 1],
\end{aligned}$$

with

$$X^N = [0,1]^3, \quad G = \{\mathbf{x} \in X^N : \sum_{i=1}^{3} x_i = 1\}.$$

The noise-free case is trivial: it was solved at the root node in all cases (AR, RAD, and RPD relaxations in all combinations with NatBds, ConvPoly1, or ConvPoly2). Similar performance was also noted for VSPODE in [114]. This problem is easy because the upper bound is within the absolute tolerance of zero, and the lower-bounding problem always returns at least zero because it is a sum of squares. Sensitivity to the choice of reference trajectory for AR relaxations is low; standard deviations in CPU time due to changes in reference trajectory are 7–16% of the mean values. The RAD relaxation method solves each instance 1.8–2.7 times faster than the average of the Singer relaxation methods for the same bounding method. All instances solved with RAD relaxations are comparable, with the standard deviation in CPU time among the different methods of about 6% of the mean value. See Table 2.3 for detailed results.

2.2.1.2 Using data with noise added from Esposito and Floudas [67]

Data with noise added were taken from [67]. There are two versions of this problem. The first uses the data for species A and B only. The second is more challenging and uses data for all three species. It is more challenging because the upper bound is larger, so the lower bound must come farther off zero. Also, since the initial condition is $(1, 0, 0)$, species C can only be formed via species B. Therefore, we expect the overestimation for species C to be at least as large as the overestimation for species B for this chemical reaction network, so fitting based on data for the concentration of species C should be at least as hard as using data on species B. For both problems, solving with the most advanced bounding and relaxation methods in dGDOpt gives performance similar or better to the results using VSPODE from [114], even after adjusting for the difference in CPU performance. See Tables 2.4 and 2.5.

2.2.1.3 Using data with noise added from [162]

The following results use the same pseudo-experimental data set and tolerance values as [162], both of which differ from those above. Even after normalizing for the differences

Table 2.3: Numerical results for reversible series reaction problem (§2.2.1.1).

bounding method					subproblem	
during	after	relaxation	upper	CPU	count	
integration	integration	method	bound	time (s)	LBP	UBP
NatBds	NatBds	AR $(\mathbf{x}^L, \mathbf{p}^L)$	$1.26{\times}10^{-6}$	0.101	1	1
NatBds	NatBds	AR $(\mathbf{x}^{mid}, \mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.101	1	1
NatBds	NatBds	AR $(\mathbf{x}^U, \mathbf{p}^U)$	$1.26{\times}10^{-6}$	0.122	1	1
NatBds	ConvPoly1	AR $(\mathbf{x}^L, \mathbf{p}^L)$	$1.26{\times}10^{-6}$	0.094	1	1
NatBds	ConvPoly1	AR $(\mathbf{x}^{mid}, \mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.087	1	1
NatBds	ConvPoly1	AR $(\mathbf{x}^U, \mathbf{p}^U)$	$1.26{\times}10^{-6}$	0.101	1	1
ConvPoly1	NatBds	AR $(\mathbf{x}^L, \mathbf{p}^L)$	$1.26{\times}10^{-6}$	0.081	1	1
ConvPoly1	NatBds	AR $(\mathbf{x}^{mid}, \mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.073	1	1
ConvPoly1	NatBds	AR $(\mathbf{x}^U, \mathbf{p}^U)$	$1.26{\times}10^{-6}$	0.096	1	1
ConvPoly1	ConvPoly1	AR $(\mathbf{x}^L, \mathbf{p}^L)$	$1.26{\times}10^{-6}$	0.079	1	1
ConvPoly1	ConvPoly1	AR $(\mathbf{x}^{mid}, \mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.072	1	1
ConvPoly1	ConvPoly1	AR $(\mathbf{x}^U, \mathbf{p}^U)$	$1.26{\times}10^{-6}$	0.093	1	1
ConvPoly2	NatBds	AR $(\mathbf{x}^L, \mathbf{p}^L)$	$1.26{\times}10^{-6}$	0.069	1	1
ConvPoly2	NatBds	AR $(\mathbf{x}^{mid}, \mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.075	1	1
ConvPoly2	NatBds	AR $(\mathbf{x}^U, \mathbf{p}^U)$	$1.26{\times}10^{-6}$	0.092	1	1
ConvPoly2	ConvPoly1	AR $(\mathbf{x}^L, \mathbf{p}^L)$	$1.26{\times}10^{-6}$	0.068	1	1
ConvPoly2	ConvPoly1	AR $(\mathbf{x}^{mid}, \mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.075	1	1
ConvPoly2	ConvPoly1	AR $(\mathbf{x}^U, \mathbf{p}^U)$	$1.26{\times}10^{-6}$	0.091	1	1
NatBds	NatBds	RAD linearized $(\mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.039	1	1
NatBds	ConvPoly1	RAD linearized $(\mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.039	1	1
ConvPoly1	NatBds	RAD linearized $(\mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.037	1	1
ConvPoly1	ConvPoly1	RAD linearized $(\mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.039	1	1
ConvPoly2	NatBds	RAD linearized $(\mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.043	1	1
ConvPoly2	ConvPoly1	RAD linearized $(\mathbf{p}^{mid})$	$1.26{\times}10^{-6}$	0.043	1	1

All instances solved to an absolute global tolerance of 10^{-4}*. CPU times are averages from 100 repetitions. Minimizer was* $(4.00, 2.00, 39.5, 19.7)$ *in all cases. No domain reduction techniques were used.*

Table 2.4: Numerical results for reversible series reaction (§2.2.1.2) using data for two species.

relaxation method	LBP count	normalized CPU time (s)
NatBds, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	96599	17010
ConvPoly1, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	69267	13447
ConvPoly2, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	17049	1732
NatBds, RPD linearized at $\mathbf{p}^{mid}$	97157	2390
ConvPoly1, RPD linearized at $\mathbf{p}^{mid}$	67789	1811
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$	13155	359
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$, GradBV	7478	192
VSPODE, fastest ε-global [114]	1622	91
VSPODE, slowest ε-global [114]	9050	211
VSPODE, fastest exact [114]	1392	89
VSPODE, slowest exact [114]	1357	272

All domain reduction techniques were used. The sensitivity right-hand sides were calculated using the built-in finite differencing scheme of CVODES for RAD and using automatic differentiation for RPD. The first seven tests are new results; the remaining entries are reproduced from [114, Table 2], with the CPU times normalized based on the PassMark benchmark from [114] being about 2.98 times slower than that of the CPU used for the new results. In all cases, the upper bound was $8.57{\times}10^{-4}$*, the relative B&B tolerance was* 10^{-3}*, and the absolute B&B tolerance was* 0*.*

Table 2.5: Numerical results for reversible series reaction (§2.2.1.2) using data for all three species.

relaxation method	LBP count	normalized CPU time (s)
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$	20503	507
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$, GradBV	15356	358
VSPODE, fastest ε-global [114]	40552	878
VSPODE, slowest ε-global [114]	10192	999
VSPODE, fastest exact [114]	4330	425
VSPODE, slowest exact [114]	7401	673

All domain reduction techniques were used. The sensitivity right-hand sides were calculated using automatic differentiation for RPD. The first two tests are new results; the remaining entries are reproduced from [114, Table 2], with the CPU times normalized based on the PassMark benchmark from [114] being about 2.98 times slower than that of the CPU used for the new results. In all cases, the upper bound was $1.59{\times}10^{-3}$*, the relative B&B tolerance was* 10^{-3}*, and the absolute B&B tolerance was* 0*.*

in processing power, all methods in dGDOpt gave significantly better CPU times than the fastest method from [162]. See Table 2.6. In all cases, we obtained a minimum at $(3.9855, 1.9823, 40.4505, 20.2308)$. All four components of the minimizer agree with [162, Section 5.5.5] to at least three significant figures. The fastest method from dGDOpt uses ConvPoly2 bounds with RAD relaxations linearized at $\mathbf{p}^{mid}$ and the GradBV method of choosing the branching variable. It was about 55 times faster than the fastest result from [162] after correcting for the difference in CPU speed using the PassMark benchmark.

2.2.2 Fed-batch control problem

This problem is from [162, Section 5.5.4]. The objective is to maximize the final-time concentration of a chemical product in an isothermal fed-batch reactor, with upper bounds constraining the final-time concentrations of two side products. The control variable is the input flow rate of one of the reactants. The flow rate is discretized into a piecewise constant function with 1, 3, 5, 7, and 9 intervals of uniform duration. For the full details of the formulation, the reader is referred to [162]. Here we formulate the problem as a minimization instead of a maximization, so the objective function has the opposite sign as that in [162]. The problem was solved with the tolerances given in Table 2.7. The B&B tolerances are identical to those used in [162]. Tighter feasibility tolerances were used in the UBP than in the LBP to ensure that the solution was truly feasible and therefore gives a valid upper bound. Looser feasibility tolerances were used in the LBP to ensure that the constraints were not overly restrictive, so that we could be sure the optimizer gave a valid lower bound. Results are given in Table 2.8. We reformulated the problem (see below) to enable use of ConvPoly bounds in addition to NatBds. For $n_p \geq 5$, RPD relaxations give consistently faster CPU times and lower node counts than IBTM. However, PRTM and PRMCTM are consistently faster than the RPD relaxations implemented in dGDOpt. We attribute this to the dependency problem in the right-hand side, which weakens the interval arithmetic and McCormick extensions used in dGDOpt, but does not weaken PRTM and PRMCTM to such a great degree. In Figure 2-1, we can see that the RPD relaxations have more favorable scaling of CPU time than IBTM [162]. The CPU time of RPD relaxations when using CPLEX to solve the lower-bounding problem appear to scale slightly better

Table 2.6: Numerical results for reversible series reaction (§2.2.1.3).

relaxation method	LBP count	normalized CPU time (s)
NatBds, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	115	8.0
ConvPoly1, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	111	6.3
ConvPoly2, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	77	4.0
NatBds, RAD linearized at $\mathbf{p}^{mid}$	495	10.6
ConvPoly1, RAD linearized at $\mathbf{p}^{mid}$	491	10.7
ConvPoly2 RAD linearized at $\mathbf{p}^{mid}$	157	4.2
NatBds, RPD linearized at $\mathbf{p}^{mid}$	163	5.7
ConvPoly1, RPD linearized at $\mathbf{p}^{mid}$	171	6.3
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$	85	11.1
NatBds, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$), GradBV	25	1.1
ConvPoly1, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$), GradBV	25	1.4
ConvPoly2, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$), GradBV	22	1.4
NatBds, RAD linearized at $\mathbf{p}^{mid}$, GradBV	41	2.7
ConvPoly1, RAD linearized at $\mathbf{p}^{mid}$, GradBV	44	1.1
ConvPoly2 RAD linearized at $\mathbf{p}^{mid}$, GradBV	23	0.6
NatBds, RPD linearized at $\mathbf{p}^{mid}$, GradBV	31	0.8
ConvPoly1, RPD linearized at $\mathbf{p}^{mid}$, GradBV	31	0.8
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$, GradBV	17	0.5
NatBds, RPD linearized at $\mathbf{p}^{mid}$, CPLEX LBP, GradBV	148	5.3
ConvPoly1, RPD linearized at $\mathbf{p}^{mid}$, CPLEX LBP, GradBV	189	5.5
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$, CPLEX LBP, GradBV	52	1.5
IBTM [162]	207	62.6
PRTM [162]	374	27.6
PRMCTM [162]	31	32.7

All domain reduction techniques were used. The sensitivity right-hand sides were calculated using the built-in finite differencing scheme of CVODES for RAD and using automatic differentiation for RPD. The first six tests are new results; the remaining entries are reproduced from [162, Table 5.13], with the CPU times normalized based on the PassMark benchmark from [162] being about 1.56 times slower than that of the CPU used for the new results. In all cases, the upper bound of -1.061523×10^{-3} *was achieved at* $(3.99, 1.98, 40.45, 20.23)$.

Table 2.7: Solver tolerances for fed-batch control problem

	absolute	relative
CVODES	10^{-9}	10^{-9}
SNOPT optimality	—	10^{-5}
SNOPT feasibility (LBP)	10^{-5}	—
SNOPT feasibility (UBP)	10^{-7}	—
B&B	10^{-3}	10^{-3}

Note: B&B tolerances are identical to those used by [162].

with n_p than the PRTM method from [162], but the PRTM and PRMCTM methods are faster in every case studied.

We reformulated this problem using total numbers of moles rather than molar concentrations, giving five state variables and two invariants:

$$\begin{aligned}
&\dot{n}_{\mathrm{A}} = -k_1 n_{\mathrm{A}} n_{\mathrm{B}}/V,\\
&\dot{n}_{\mathrm{B}} = -(n_{\mathrm{B}}/V)(k_1 n_{\mathrm{A}} + 2k_2 n_{\mathrm{B}}) + u c_{\mathrm{B},in},\\
&\dot{n}_{\mathrm{C}} = k_1 n_{\mathrm{A}} n_{\mathrm{B}}/V,\\
&\dot{n}_{\mathrm{D}} = k_2 n_{\mathrm{B}}^2/V,\\
&\dot{V} = u,\\
&k_1 = 0.053, \quad k_2 = 0.128,\\
&\mathbf{x}_0 \equiv (n_{\mathrm{A},0}, n_{\mathrm{B},0}, n_{\mathrm{C},0}, n_{\mathrm{D},0}, V_0) \equiv (0.72, 0.05, 0, 0, 1),\\
&X^N \equiv [0, 0.72] \times [0, 0.3] \times [0, 0.3] \times [0, 0.15] \times [0, 1.05],\\
&G \equiv \{\mathbf{z} \in X^N : n_{\mathrm{A}} + n_{\mathrm{C}} = 0.72 \text{ and } n_{\mathrm{B}} + n_{\mathrm{C}} + 2n_{\mathrm{D}} - 5V = -4.95\},\\
&P \equiv [0, 0.001]^{n_p},\\
&t \in [t_0, t_f] \equiv [0, 50],
\end{aligned} \tag{2.3}$$

where u is the inlet flow rate given by a piecewise constant control parameterization with time intervals of uniform duration.

The invariant quantities $n_{\mathrm{A}} + n_{\mathrm{C}}$ and $n_{\mathrm{B}} + n_{\mathrm{C}} + 2n_{\mathrm{D}} - 5V$ were obtained by writing an

Table 2.8: Numerical results for fed-batch control problem (§2.2.2).

	relaxation	LBP	normalized CPU
n_p	method	count	time (s)
1	RPD ($\mathbf{p}^{mid}$), reform., GradBV, NatBds	5	3.11
1	RPD ($\mathbf{p}^{mid}$), reform., GradBV, ConvPoly1	5	0.41
1	RPD ($\mathbf{p}^{mid}$), reform., GradBV, ConvPoly2	5	0.43
1	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, NatBds	5	2.84
1	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly1	5	0.36
1	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly2	5	0.38
1	IBTM [162]	5	0.06
1	PRTM [162]	1	0.03
1	PRMCTM [162]	1	0.04
3	RPD ($\mathbf{p}^{mid}$), reform., GradBV, NatBds	22	19.8
3	RPD ($\mathbf{p}^{mid}$), reform., GradBV, ConvPoly1	22	18.5
3	RPD ($\mathbf{p}^{mid}$), reform., GradBV, ConvPoly2	22	20.0
3	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, NatBds	21	16.8
3	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly1	21	14.6
3	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly2	21	13.6
3	IBTM [162]	215	4.54
3	PRTM [162]	3	0.21
3	PRMCTM [162]	3	0.29
5	RPD ($\mathbf{p}^{mid}$), reform., GradBV, NatBds	486	172
5	RPD ($\mathbf{p}^{mid}$), reform., GradBV, ConvPoly1	480	157
5	RPD ($\mathbf{p}^{mid}$), reform., GradBV, ConvPoly2	458	166
5	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, NatBds	110	67.6
5	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly1	102	65.2
5	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly2	104	62.2
5	IBTM [162]	65, 043	3, 132.
5	PRTM [162]	27	6.2
5	PRMCTM [162]	27	6.8
7	RPD ($\mathbf{p}^{mid}$), reform., GradBV, NatBds	9, 104	3, 893.
7	RPD ($\mathbf{p}^{mid}$), reform., GradBV, ConvPoly1	9, 260	3, 989.
7	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, NatBds	1,380	793.
7	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly1	1,348	787.
7	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly2	1,358	848.
7	IBTM [162]	$>$ 250, 000	$>$64, 000.
7	PRTM [162]	179	218.
7	PRMCTM [162]	73	65.
9	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, NatBds	22,704	15, 904.
9	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly1	22,892	15, 652.
9	RPD ($\mathbf{p}^{mid}$), reform., CPLEX LBP, GradBV, ConvPoly2	23,894	17, 089.
9	PRTM [162]	1,007	6, 189.
9	PRMCTM [162]	209	895.

Tests 1 & 2 and probing for domain reduction were used in all cases. For the results from dGDOpt, ConvPoly1 bounds were always used for post-integration pruning; integration bounding method is noted. The sensitivity right-hand sides were calculated using the built-in finite differencing scheme of CVODES. The RPD results in each block are new results; the last three results in each block are reproduced from [162], with CPU times normalized based on the CPU in [162] having a PassMark benchmark about 1.56 times slower than the CPU used here.

Table 2.9: Solutions to flow control problem (§2.2.2).

n_p	upper bound	example minimizer
1	$-4.857{\times}10^{-2}$	$(1.0147{\times}10^{-4})$
3	$-4.966{\times}10^{-2}$	$(0, 2.424{\times}10^{-4}, 1.193{\times}10^{-4})$
5	$-4.967{\times}10^{-2}$	$(0, 5.504{\times}10^{-5}, 2.370{\times}10^{-4}, 2.234{\times}10^{-4}, 8.751{\times}10^{-5})$
7	$-4.970{\times}10^{-2}$	$(0, 0, 1.442{\times}10^{-4}, 2.219{\times}10^{-4}, 1.870{\times}10^{-4}, 2.398{\times}10^{-4}, 5.243{\times}10^{-5})$
9	$-4.971{\times}10^{-2}$	$(0, 0, 0, 2.342{\times}10^{-4}, 1.949{\times}10^{-4}, 2.079{\times}10^{-4}, 1.873{\times}10^{-4}, 2.493{\times}10^{-4}, 1.343{\times}10^{-5})$

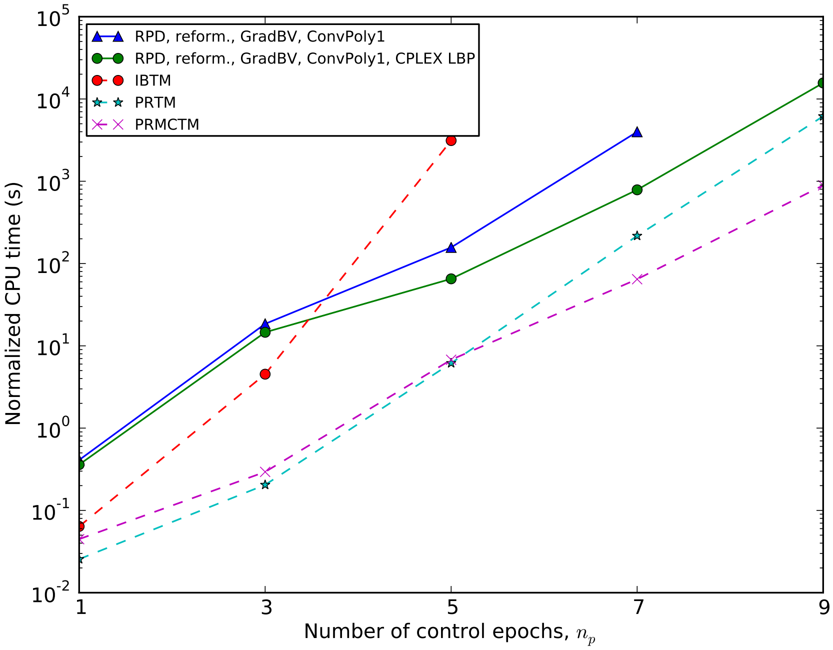


Figure 2-1: CPU time for methods from the present work scale significantly better with n_p than previously-reported results using IBTM for the fed-batch control problem. The two solid lines connect CPU times measured in the current work; dashed lines connect CPU times reported in [162]. PRTM and PRMCTM methods were faster than the present work for all values of n_p tested, however, with CPLEX for the LBP, our software appears to scale better than PRTM and PRMCTM. CPU times from [162] were normalized based on the PassMark benchmarks for the respective CPUs. All domain reduction options were used in all cases shown.

augmented stoichiometry matrix for the flow system

$$\mathbf{S}_{\text{aug}} = \begin{bmatrix} -1 & 0 & 0 \\ -1 & -2 & c_{\text{B},in} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and computing the null space of $\mathbf{S}_{\text{aug}}^{\text{T}}$. The first two columns of $\mathbf{S}_{\text{aug}}$ correspond to the reactions A + B → C and 2B → D; the third column corresponds to the inflow containing species B at concentration $c_{\text{B},in} = 5$. The idea of using an augmented stoichiometry matrix to compute invariants for the system was inspired by [5, §3.1.4].

2.2.3 Singular control problem

This three-state, one-control singular control problem from [120] has also been solved in [66, 116, 187].

$$\min_{\mathbf{p}} \int_0^1 \left(x_1^2 + x_2^2 + 0.0005(x_2 + 16t - 8 - 0.1x_3u^2)^2\right) dt$$

$$\text{s.t. } u(t) = \begin{cases} p_1 & \text{if } t \in [t_0, (t_f - t_0)/n_{\text{control}} + t_0), \\ \vdots & \\ p_{n_{\text{control}}} & \text{if } t \in [(n_{\text{control}} - 1)(t_f - t_0)/n_{\text{control}} + t_0, t_f], \end{cases}$$

$$\mathbf{p} \in [-4, 10]^{n_{\text{control}}},$$

where the state variables $\mathbf{x}$ are given by the solution of the initial value problem

$$\begin{aligned}
&\dot{x}_1 = x_2,\\
&\dot{x}_2 = -x_3 u + 16t - 8,\\
&\dot{x}_3 = u,\\
&\mathbf{x}(t_0) = (0, -1, -\sqrt{5}),\\
&t \in [t_0, t_f] = [0, 1],
\end{aligned}$$

where

$$X^N = G = \mathbb{R}^3.$$

This problem is not derived from a physical system and no natural bounding set exists. Using AR relaxations to solve this problem, the amount of CPU time and number of subproblems required are very sensitive to the reference trajectory, varying by a factor of about 30 under changes in reference trajectory (Table 2.10). For this particular problem, RPD relaxations yield consistently slower CPU times than RAD relaxations. This is because the RPD relaxations offer no benefit in tightness since for each i, the time derivative $\dot{x}_i$ does not depend on the current value of state x_i, so the flattening step in the computation of RPD ($\mathcal{R}_i^{cv/cc}$ operator in [170, §7.6.3]) has no benefit. In our tests, slightly fewer lower-bounding problems are required for RPD as compared to RAD because our implementation of RPD uses event detection to ensure the state relaxations always stay inside the state bounds. For some nodes in the B&B tree using RAD relaxations, the relaxations leave the bounds, leading to larger values of the state variables in the integrator and more numerical integration failures. In the end, RAD is still significantly faster because the average cost of each lower-bounding problem is so much lower than that for RPD and the relaxations of the objective function are of equal strength except when there are numerical integration failures. See Table 2.11. We further tailored the optimization methods to this problem by solving using RAD without performing the flattening step in the state bounding system (NaïveBds). Again, since no $\dot{x}_i$ depends on x_i, this does not worsen the bounds, but it decreases the cost of evaluating the right-hand side of the bounding system.

Table 2.10: Numerical results for Singular Control problem (§2.2.3) using AR relaxations are highly sensitive to the reference trajectory.

relaxation method	LBP count	normalized CPU time (s)
NatBds, AR $(\mathbf{x}^{*,L}, \mathbf{p}^{*,L})$	20953	4106.0
NatBds, AR $(\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid})$	607	184.0
NatBds, AR $(\mathbf{x}^{*,U}, \mathbf{p}^{*,U})$	19735	3812.0
NatBds, RAD linearized at $\mathbf{p}^{mid}$	633	124.0

All instances solved to a global absolute tolerance of 10^{-3} *with* $n_p = 3$. *Natural bounds were used in all cases, with bounds effectively* $(-\infty, +\infty)$ *since the system has no affine invariants and no* a priori *known bounds. Tests 1 and 2 as well as probing for domain reduction were enabled. In all cases, the upper bound was 0.1475.*

The singular control problem was solved with 1, 2, 3, 4, and 5 piecewise constant control epochs to examine the scaling with CPU time. For small numbers of control epochs, Lin and Stadtherr [116] and Sahlodin [162] solved the problem more quickly than we have here, but as the number of control epochs increases, dGDOpt solves the problem more quickly than some of the other methods. For $n_p = 1$ and $n_p = 2$, RAD and RPD relaxations are consistently slower than the methods from [162]. However, for $n_p \geq 3$, RAD yields faster CPU times than IBTM even after adjusting for differences in CPU performance. For $n_p = 5$, RAD gives faster global optimization than all three methods from [162] after normalizing for CPU performance. Figure 2-2 makes it clear that RAD with NatBds scales more favorably than the methods from [116, 162]—the only other methods reported to have solved this problem to guaranteed global optimality.

2.2.4 Denbigh problem

This problem is taken from [162, Section 5.5.3]. Sahlodin adapted the problem slightly from [58]. We solved an equivalent problem to that solved in Sahlodin's code, which differs slightly from what is printed in [162]. We used absolute and relative tolerances on the function value of 10^{-3}, in agreement with [162]. In personal communication with Dr. Sahlodin, we established that the formulation used to generate the results results in [162] is:

$$\min_{\mathbf{p} \in P} -x_2(t_f)$$

Table 2.11: Numerical results for Singular Control problem (§2.2.3) with 1 to 5 control epochs.

relaxation		upper		LBP	normalized
method	probing?	bound	minimizer	count	CPU time (s)
AR $(\mathbf{x}^{mid*}, \mathbf{p}^{mid})$	yes	0.4965	(4.071)	15	1.07
RAD, NaïveBds	yes	0.4965	(4.071)	13	0.37
RAD	no	0.4965	(4.071)	13	0.29
RAD	yes	0.4965	(4.071)	13	0.19
RAD, GradBV	yes	0.4965	(4.071)	13	0.29
RPD	yes	0.4965	(4.071)	13	4.54
IBTM	yes	0.4965	(4.071)	5	0.06
PRTM	yes	0.4965	(4.071)	3	0.03
PRMCTM	yes	0.4965	(4.071)	3	0.04
AR $(\mathbf{x}^{mid*}, \mathbf{p}^{mid})$	yes	0.2771	$(5.575, -4.000)$	43	7.62
RAD, NaïveBds	yes	0.2771	$(5.575, -4.000)$	49	2.16
RAD	no	0.2771	$(5.550, -4.000)$	39	4.32
RAD	yes	0.2771	$(5.575, -4.000)$	39	3.21
RAD, GradBV	yes	0.2771	$(5.575, -4.000)$	32	1.77
RPD	yes	0.2771	$(5.575, -4.000)$	39	3.86
IBTM	yes	0.2771	$(5.575, -4.000)$	55	1.28
PRTM	yes	0.2771	$(5.575, -4.000)$	27	0.48
PRMCTM	yes	0.2771	$(5.575, -4.000)$	27	0.63
AR $(\mathbf{x}^{mid*}, \mathbf{p}^{mid})$	yes	0.1475	$(8.002, -1.944, 6.042)$	607	139
RAD, NaïveBds	yes	0.1475	$(8.002, -1.944, 6.042)$	591	31
RAD	no	0.1475	$(7.972, -1.916, 6.061)$	597	147
RAD	yes	0.1475	$(8.002, -1.944, 6.042)$	591	34
RAD, GradBV	yes	0.1475	$(8.002, -1.944, 6.042)$	484	24
RPD	yes	0.1475	$(8.002, -1.944, 6.042)$	603	51
IBTM	yes	0.1475	$(8.002, -1.944, 6.042)$	1,367	45
PRTM	yes	0.1475	$(8.002, -1.944, 6.042)$	445	17
PRMCTM	yes	0.1475	$(8.002, -1.944, 6.042)$	443	22
AR $(\mathbf{x}^{mid*}, \mathbf{p}^{mid})$	yes	0.1237	$(9.789, -1.199, 1.256, 6.356)$	8,547	$3,038$
RAD, NaïveBds	yes	0.1237	$(9.789, -1.199, 1.256, 6.356)$	8,919	641
RAD	no	0.1237	$(9.789, -1.205, 1.255, 6.371)$	9,435	$1,079$
RAD	yes	0.1237	$(9.789, -1.199, 1.256, 6.356)$	8,919	726
RAD, GradBV	yes	0.1237	$(9.789, -1.199, 1.256, 6.356)$	6,578	572
RPD	yes	0.1237	$(9.789, -1.199, 1.256, 6.356)$	9,081	656
IBTM	yes	0.1238	$(9.789, -1.200, 1.257, 6.356)$	28,809	$1,419$
PRTM	yes	0.1238	$(9.789, -1.200, 1.257, 6.356)$	5,159	474
PRMCTM	yes	0.1238	$(9.789, -1.200, 1.257, 6.356)$	5,137	553
AR $(\mathbf{x}^{mid*}, \mathbf{p}^{mid})$	yes	0.1236	$(10.000, 1.494, -0.814, 3.352, 6.154)$	92,183	$42,037$
RAD, NaïveBds	yes	0.1236	$(10.000, 1.494, -0.814, 3.352, 6.154)$	83,469	$7,027$
RAD	no	0.1236	$(9.994, 1.554, -0.938, 3.479, 6.063)$	172,675	$20,474$
RAD	yes	0.1236	$(10.000, 1.494, -0.814, 3.352, 6.154)$	83,469	$8,353$
RAD, GradBV	yes	0.1236	$(10.000, 1.498, -0.817, 3.330, 6.184)$	59,364	$5,666$
RPD	yes	0.1236	$(10.000, 1.494, -0.814, 3.352, 6.154)$	83,507	$13,275$
IBTM	yes	0.1236	$(10.000, 1.494, -0.814, 3.354, 6.151)$	504,827	$25,067$
PRTM	yes	0.1236	$(10.000, 1.494, -0.814, 3.354, 6.151)$	54,617	$11,890$
PRMCTM	yes	0.1236	$(10.000, 1.494, -0.814, 3.354, 6.151)$	55,107	$13,792$

All instances solved to global absolute and relative tolerances of 10^{-3}, just as in [162]. NatBds were used unless otherwise noted, with bounds effectively $(-\infty, +\infty)$ for all states, since the system has no affine invariants and no natural bounds. Tests 1 & 2 for domain reduction were always enabled. All RAD and RPD relaxations were linearized at $\mathbf{p}^{mid}$. CPU time data for IBTM, PRTM, and PRMCTM have been normalized from [162, Tables 5.3 and 5.4], which used a CPU with a PassMark benchmark about 1.56 times slower than the CPU used here.

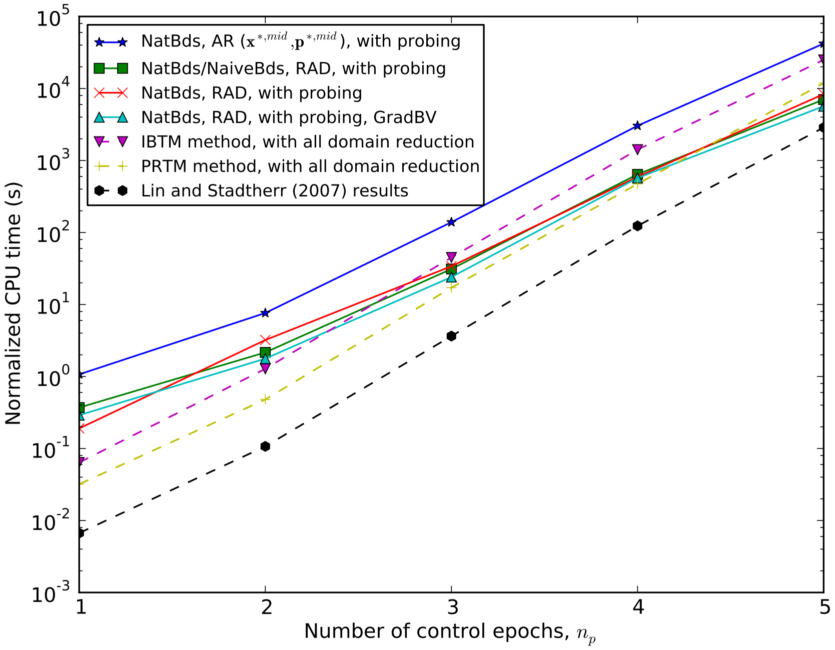


Figure 2-2: CPU time for some methods from dGDOpt scale better with n_p than previously-reported results for the singular control problem. The three solid lines connect CPU times measured in the current work; dashed lines connect CPU times reported in [162]; a dot-dashed line connects CPU times from [116]. All instances were solved to a global absolute tolerance of 10^{-3}. CPU times from [162] and [116] were normalized based on the PassMark benchmarks for the respective CPUs.

where $\mathbf{x}(t_f)$ is given by the solution of the IVP:

$$\begin{aligned}
&\dot{x}_1 = -k_1 x_1^2,\\
&\dot{x}_2 = k_1 x_1^2 - k_2 x_2,\\
&\mathbf{x}(t_0) = (1,0),\\
&t \in [t_0, t_f] = [0,10],
\end{aligned}$$

where

$$\begin{aligned}
&k_i = a_i \exp\left(\frac{-b_i p}{298R}\right),\ i = 1,2,\\
&\mathbf{p} \in [298/423, 1]^{n_p},\\
&a_1 = 4.0{\times}10^3,\ a_2 = 6.2{\times}10^4,\\
&b_1/R = 2.5{\times}10^3,\ b_2/R = 5.0{\times}10^3,
\end{aligned}$$

with $n_p = 1,\ldots,4$ for different numbers of piecewise constant control parameters on a uniform time grid. In particular, note that the definitions for k_i and a_2 used here differ from those printed in [162], but they are the definitions actually used to generate the results in [162] and in the present work. We added an additional state to the system, such that the solution obeys an affine invariant and the natural bounds below:

$$\begin{aligned}
&\dot{x}_1 = -k_1 x_1^2,\\
&\dot{x}_2 = k_1 x_1^2 - k_2 x_2,\\
&\dot{x}_3 = k_2 x_2,\\
&\mathbf{x}(t_0) = (1,0,0),\\
&X^N \equiv [0,1] \times [0,1] \times [0,1],\\
&G \equiv \{\mathbf{z} \in X^N : x_1 + x_2 + x_3 = x_{1,0} + x_{2,0} + x_{3,0} = 1\},\\
&t \in [t_0, t_f] = [0,10].
\end{aligned}$$

CPU times and LBP counts are given in Table 2.12; solutions are given in Table 2.13.

The methods from [162] give about an two to five times faster CPU times than the fastest methods implemented for this work.

Table 2.12: Numerical results for Denbigh problem (§2.2.4) with 1 to 4 control epochs.

	relaxation	LBP	normalized
n_p	method	count	CPU time (s)
1	NatBds, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	33	2.5
1	ConvPoly1, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	29	5.2
1	ConvPoly2, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	27	2.3
1	NatBds, RAD	97	3.2
1	ConvPoly1, RAD	67	2.3
1	ConvPoly2, RAD	53	2.3
1	NatBds, RPD	31	6.1
1	ConvPoly1, RPD	29	3.3
1	ConvPoly2, RPD	23	1.7
1	NatBds, RPD, CPLEX LBP	35	0.9
1	ConvPoly1, RPD, CPLEX LBP	33	0.7
1	ConvPoly2, RPD, CPLEX LBP	29	0.7
1	IBTM	7	0.3
1	PRTM	7	0.3
1	PRMCTM	7	0.6
2	NatBds, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	213	29.2
2	ConvPoly1, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	206	32.2
2	ConvPoly2, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	187	27.2
2	NatBds, RAD	916	46.7
2	ConvPoly1, RAD	886	46.6
2	ConvPoly2, RAD	870	55.0
2	NatBds, RPD	195	49.2
2	ConvPoly1, RPD	190	38.6
2	ConvPoly2, RPD	179	21.4
2	NatBds, RPD, CPLEX LBP	220	5.1
2	ConvPoly1, RPD, CPLEX LBP	224	5.2
2	ConvPoly2, RPD, CPLEX LBP	208	4.8
2	IBTM	31	2.7
2	PRTM	27	1.8
2	PRMCTM	27	3.2
3	NatBds, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	10328	2697.0
3	ConvPoly1, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	10594	2793.0
3	ConvPoly2, AR ($\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid}$)	1768	462.0
3	NatBds, RAD	8004	628.0
3	ConvPoly1, RAD	8056	646.0
3	ConvPoly2, RAD	7652	701.0
3	NatBds, RPD	1562	928.0

3	ConvPoly1, RPD	1448	344.0	▭
3	ConvPoly2, RPD	882	153.0	▯
3	NatBds, RPD, CPLEX LBP	1640	40.1	▯
3	ConvPoly1, RPD, CPLEX LBP	1508	38.6	▯
3	ConvPoly2, RPD, CPLEX LBP	1482	39.7	▯
3	IBTM	111	10.8	|
3	PRTM	93	9.9	|
3	PRMCTM	93	16.1	|
4	NatBds, RPD, CPLEX LBP	9956	306.0	▭
4	ConvPoly1, RPD, CPLEX LBP	8940	300.0	▭
4	ConvPoly2, RPD, CPLEX LBP	8724	303.0	▭
4	IBTM	411	63.5	▯
4	PRTM	303	55.4	▯
4	PRMCTM	295	81.4	▯

All instances solved to global absolute and relative tolerances of 10^{-3}, just as in [162]. Tests 1 & 2 and probing for domain reduction were always enabled. GradBV was used in all cases except IBTM, PRTM, and PRMCTM. All RAD and RPD relaxations were linearized at $\mathbf{p}^{mid}$. CPU time data for IBTM, PRTM, and PRMCTM have been normalized from [162, Tables 5.3 and 5.4], which used a CPU with a PassMark benchmark about 1.56 times slower than the CPU used here. For each value of n_p, all solvers and options gave the same solutions to at least 3 significant figures (Table 2.13).

2.2.5 Oil shale pyrolysis optimal control problem

This two-state, one-control optimal control problem has been studied by several authors [44, 120, 157, 187]. Here, piecewise constant control parameterization is used with a uniform time grid in one to three control epochs ($n_p = 1, 2, 3$). The problem is:

$$\min_{\mathbf{p}} -x_2(t_f),$$

Table 2.13: Solutions to Denbigh problem (§2.2.4).

n_p	upper bound	minimizer
1	-0.8811	(0.977)
2	-0.8813	$(0.969, 0.985)$
3	-0.8814	$(0.963, 0.983, 0.986)$
4	-0.8815	$(0.958, 0.980, 0.985, 0.987)$

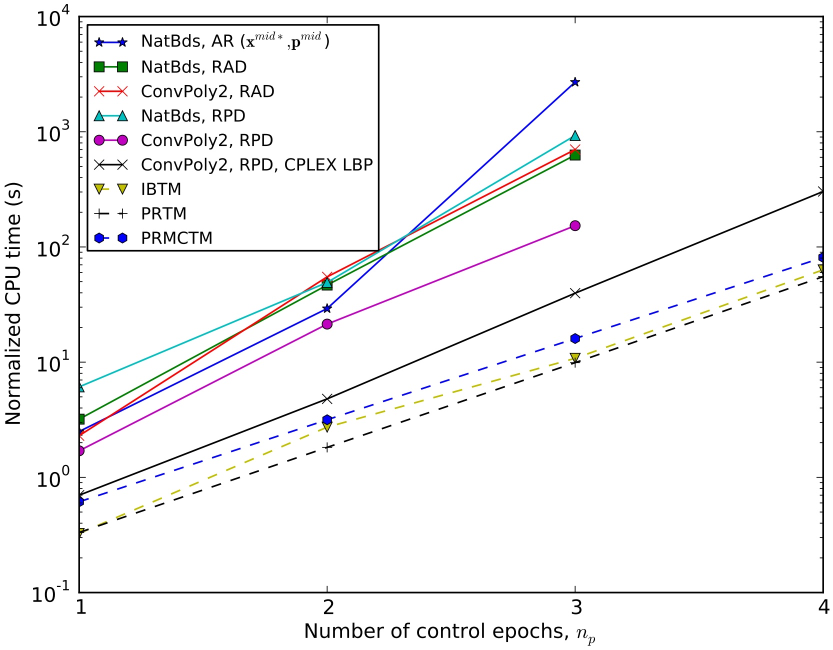


Figure 2-3: CPU times using RPD in dGDOpt scale similarly with n_p as previously-reported results for the Denbigh problem. The six solid lines connect CPU times measured in the current work; dashed lines connect CPU times reported in [162]. All instances were solved to a global absolute tolerance of 10^{-3}. CPU times from [162] were normalized based on the PassMark benchmarks for the respective CPUs.

Table 2.14: Values of constants in Oil Shale Pyrolysis problem

i	a_i	b_i/R
1	7.044482745×10^{3}	-10215.4
2	3.401270608×10^{10}	-18820.5
3	1.904365858×10^{10}	-17008.9
4	1.390021558×10^{8}	-14190.8
5	9.770027258×10^{8}	-15599.8

where $x_2(t_f)$ is given by the solution of the initial value problem:

$$\begin{aligned}
&\dot{x}_1 = -x_1(k_1 + k_3x_2 + k_4x_2 + k_5x_2), \quad \forall t \in (t_0, t_f] \\
&\dot{x}_2 = x_1(k_1 + k_3x_2) - k_2x_2, \quad \forall t \in (t_0, t_f] \\
&k_i = a_i \exp\left[\frac{-b_i/R}{698.15 + 50u}\right], \quad i = 1, \dots, 5, \\
&\mathbf{x}(t_0) = (1, 0), \\
&t \in [t_0, t_f] \equiv [0, 10],
\end{aligned}$$

where u is given by a piecewise constant control parameterization on a uniform grid with n_p epochs, $\mathbf{p} \in [0,1]^{n_p}$, and

$$X^N = [0,1]^2, \quad G = X^N.$$

For this problem, there are no affine invariants in the original formulation so only natural bounds can be used in the original formulation. However, an additional state can be added, yielding one affine invariant. Specifically, the additional state is:

$$\dot{x}_3 = x_2(k_2 + (k_4 + k_5)x_1), \qquad x_3(t_0) = 0, \tag{2.4}$$

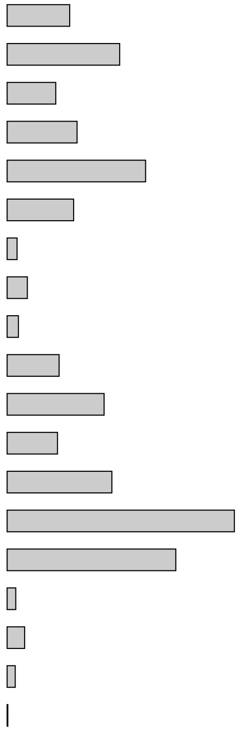
with the new sets

$$X^N = [0,1]^3, \quad G = \{\mathbf{z} \in X^N : z_1 + z_2 + z_3 = 1\}.$$

With NatBds on the original formulation, RPD relaxations gave six to seven times faster CPU times than RAD and AR for $n_p = 2$. Within tests of each relaxation method, ConvPoly2 gave slightly faster times than NatBds, while ConvPoly1 gave slower times. The

Table 2.15: Numerical results for Oil Shale Pyrolysis problem (§2.2.5) with $n_p = 2$

relaxation method	LBP count	normalized CPU time (s)
NatBds, AR $(\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid})$	5001	840
ConvPoly1, AR $(\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid})$	4967	1511
ConvPoly2, AR $(\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid})$	2047	653
NatBds, RAD linearized at $\mathbf{p}^{mid}$	14599	938
ConvPoly1, RAD linearized at $\mathbf{p}^{mid}$	14593	1858
ConvPoly2, RAD linearized at $\mathbf{p}^{mid}$	6849	893
NatBds, RPD linearized at $\mathbf{p}^{mid}$	4639	133
ConvPoly1, RPD linearized at $\mathbf{p}^{mid}$	4607	271
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$	2037	151
NatBds, AR $(\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid})$, GradBV	4550	697
ConvPoly1, AR $(\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid})$, GradBV	4502	1301
ConvPoly2, AR $(\mathbf{x}^{*,mid}, \mathbf{p}^{*,mid})$, GradBV	2272	676
NatBds, RAD linearized at $\mathbf{p}^{mid}$, GradBV	18242	1406
ConvPoly1, RAD linearized at $\mathbf{p}^{mid}$, GradBV	19060	3050
ConvPoly2, RAD linearized at $\mathbf{p}^{mid}$, GradBV	13804	2264
NatBds, RPD linearized at $\mathbf{p}^{mid}$, GradBV	4254	116
ConvPoly1, RPD linearized at $\mathbf{p}^{mid}$, GradBV	4364	235
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$, GradBV	1836	109
VSPODE [116]	178	9



All instances, including the literature result, were solved to a global absolute tolerance of 10^{-3}*. The dynamic system solved using natural bounds had 2 states; that solved using ConvPoly bounds had 3 states. An extra state was added to create the affine invariant to enforce in the ConvPoly runs. Tests 1 and 2 and probing for domain reduction were enabled. All results except that from VSPODE are from the present study. The results from VSPODE were normalized based on the PassMark benchmark being 2.98× faster for one core of the present CPU compared to that in [116]. In all cases, the minimum function value of* -0.351 *was attained at* $(0.431, 0.000)$*.*

normalized CPU time for VSPODE is about 10 times faster than that for dGDOpt on this problem. See Table 2.15.

2.2.6 Pharmaceutical reaction model

This seven-state, five-parameter problem is from the Novartis-MIT Center for Continuous Manufacturing. It is an unweighted least-squares parameter estimation problem where the state variables $\mathbf{x}$ are given by the solution of the initial value problem

$$\begin{aligned}
&\dot{\mathbf{x}} = \mathbf{S}\mathbf{r},\\
&\mathbf{r} = (k_1 x_1, k_2 x_4, k_3 x_1, k_4 x_4, k_5),\\
&k_i = \exp(p_i),\\
&\mathbf{p} \in [-2, 1]^4 \times [-6, -3],\\
&\mathbf{x}(t_0) = (0.792302, 3.942141, 0, 0, 0, 0, 0),\\
&t \in [t_0, t_f] = [0, 5./3.],
\end{aligned}$$

where the stoichiometry matrix $\mathbf{S}$ is given by

$$\mathbf{S} = \begin{bmatrix} -1 & 1 & -1 & 0 & -0.5 \\ -1 & 1 & 0 & 0 & -1.5 \\ 1 & -1 & 0 & -1 & -0.5 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

Table 2.16: Experimental data for pharmaceutical reaction model (§2.2.6).

t (hours)	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	0.792302	3.942141	0.000000	0.000000	0.000000	0.000000	0.000000
1/6	0.594226	3.751241	0.186349	0.003565	0.000657	0.000000	0.000657
2/6	0.294261	3.458084	0.466507	0.011251	0.004199	0.006259	0.004199
3/6	0.181754	3.345181	0.570457	0.015450	0.007368	0.005625	0.007368
4/6	0.134612	3.298436	0.611340	0.017510	0.009904	0.005467	0.009904
5/6	0.115835	3.279183	0.626711	0.018540	0.011805	0.004675	0.011805
1	0.108070	3.270745	0.630672	0.019570	0.014103	0.004595	0.014103
7/6	0.104663	3.268447	0.628295	0.020204	0.016797	0.005150	0.016797
8/6	0.103950	3.267100	0.625522	0.020758	0.019174	0.005150	0.019174
9/6	0.104188	3.268685	0.620214	0.021154	0.021392	0.005467	0.021392
10/6	0.103792	3.266347	0.617758	0.021788	0.024165	0.005705	0.024165

All concentrations in mol/L.

and

$$X^N = [-0.0242, 0.792302] \times [2.3091, 3.942142] \times [-0.0242, 0.792302] \times [-0.0242, 0.792302]$$
$$\times [-0.0242, 0.792302] \times [-0.0242, 0.792302] \times [-0.0242, 0.792302],$$

$$G = \left\{ \mathbf{z} \in X^N : \begin{array}{l} z_1 + z_3 + z_4 + z_6 + z_7 = x_{1,0} + x_{3,0} + x_{4,0} + x_{6,0} + x_{7,0}, \\ z_2 + z_3 + z_4 + 2z_7 = x_{2,0} + x_{3,0} + x_{4,0} + 2x_{7,0}, \text{ and} \\ z_5 - z_7 = x_{5,0} - x_{7,0} \end{array} \right\}.$$

Dr. Sahlodin gave us his GDOPT code used for [162]; we solved this problem using his software in addition to our own. Both codes gave the same minimizer and upper bound, but our fastest method solved the problem over 15 times faster than the fastest method from [162].

Figure 2-4 illustrates the impact of using the GradBV rule to select the branching variable rather than the AbsDiamBV rule.

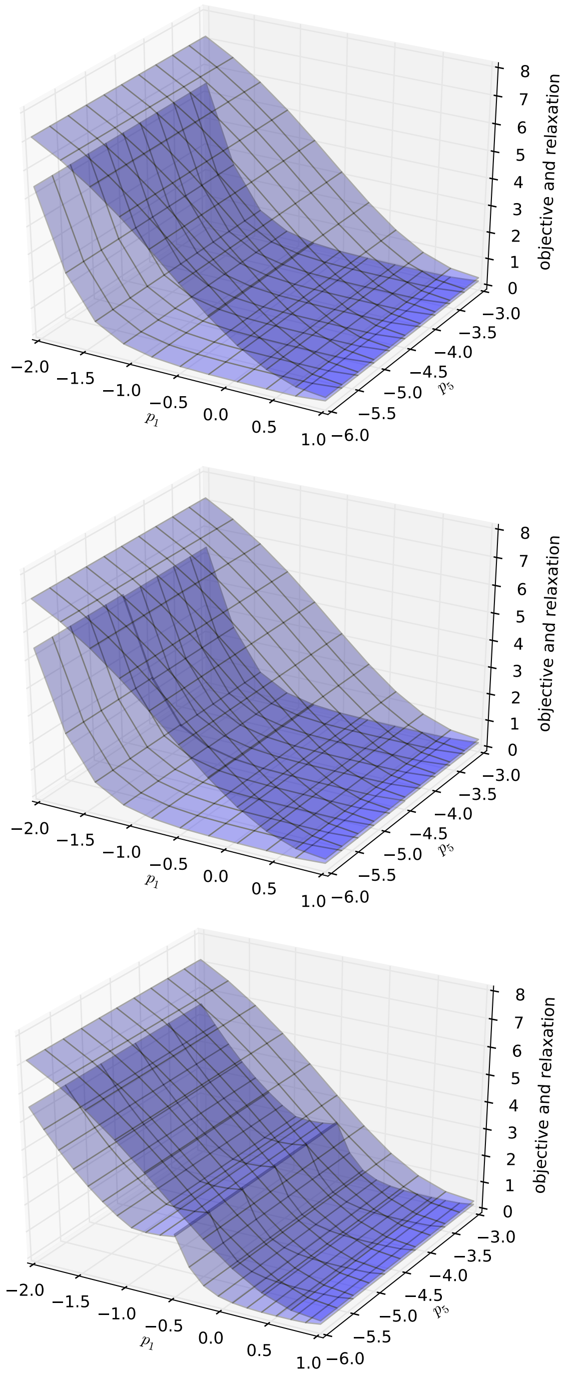


Figure 2-4: Objective function and its convex relaxation generated plotted versus p_1 and p_5 for pharmaceutical reaction model (§2.2.6) on (top) root node, (middle) two child nodes when partitioned at the plane $p_5 = 0.5(p_5^L + p_5^U)$, and (bottom) two child nodes partitioned at the plane $p_1 = 0.5(p_1^L + p_1^U)$. Partitioning at the midpoint of p_5 does not visibly improve the relaxations and does not allow eliminating any of the search space from consideration. Partitioning at the midpoint of p_1 visibly improves the relaxations and is sufficient to eliminate half of the search space from consideration. Given the choice between only p_1 and p_5 as branching variables, GradBV would select p_1 for partitioning (thus eliminating half of the search space) whereas AbsDiamBV could select p_5 (thus not eliminating any of the search space) because both p_1 and p_5 have equal absolute diameters at the root node.

Table 2.17: Numerical results for pharmaceutical reaction network (§2.2.6).

relaxation method	LBP count	CPU time (s)
NatBds, RPD linearized at $\mathbf{p}^{mid}$	75030	3534
ConvPoly1, RPD linearized at $\mathbf{p}^{mid}$	75030	3622
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$	75548	4456
NatBds, RPD linearized at $\mathbf{p}^{mid}$, GradBV	260	17
ConvPoly1, RPD linearized at $\mathbf{p}^{mid}$, GradBV	260	16
ConvPoly2, RPD linearized at $\mathbf{p}^{mid}$, GradBV	262	21
IBTM	705	266
PRTM	705	266
PRMCTM	569	279

All domain reduction techniques were used. Probing used the linearized states, so only a single integration was required for each lower bounding problem and the subsequent $2n_p$ *probing problems. The absolute and relative B&B tolerances were* 10^{-2}. *An upper bound of 0.1205 at* $(0.811, 0.000, -1.756, 0.000, -4.31)$ *was achieved in all cases. All other settings for the code from [162] were left at defaults.*

2.2.7 Discretized PDE problem to show scaling with n_x

The next problem is inspird by a discretized PDE with varying numbers of states, n_x, to elucidate the scaling of the methods with n_x. We simulated the dynamic system

$$\begin{aligned}
\dot{x}_1 &= \frac{-cn_x}{\zeta_{\max} - \zeta_{\min}}(x_1), \\
\dot{x}_i &= \frac{-cn_x}{\zeta_{\max} - \zeta_{\min}}(x_i - x_{i-1}), \quad i = 2, \dots, n_x - 1, \\
\dot{x}_{n_x} &= \frac{-cn_x}{\zeta_{\max} - \zeta_{\min}}(-x_{n_x-1}),
\end{aligned} \tag{2.5}$$

with the initial condition $\mathbf{x}(t_0) = (1, 0, \dots, 0)$. For each value of n_x tested, we simulated data for the system using $\frac{c}{\zeta_{\max} - \zeta_{\min}} = 5.0$ in Matlab using ode15s with absolute and relative tolerances of 10^{-9}, then added the pseudorandom noise in Table 2.18 to the simulated data for (x_{n_x-1}, x_{n_x}) to form the pseudoexperimental data $(\widehat{x}_{n_x-1}, \widehat{x}_{n_x})$. Finally, we formed an unweighted least-squares parameter estimation problem using those 20 data points, letting

$p \in P \equiv [0, 10]$. The system obeys a natural bound and an affine invariant:

$$X^N = [0, 1]^{n_x},$$
$$G = \{\mathbf{z} \in X^N : \sum_i z_i = 1\},$$

which we exploited where noted. The problem was solved using CPLEX for the lower-bounding problem in all cases.

The problem was solved with up to 41 state variables. Relaxations from RAD yielded an initial large increase in LBP count with n_x, then plateaued around 600 LBPs. Relaxations from RPD and AR yielded a more gradual increase with up to about 150 LBPs. CPU time scaled the most favorably with ConvPoly1/RAD and NatBds/RAD, followed by ConvPoly1/AR and NatBds/AR, then RPD with all three bounding methods. ConvPoly2/RAD and ConvPoly2/AR gave the least favorable scaling of CPU time with n_x. This is expected since the number of operations for the ConvPoly2 bounding method scales as n_x^3.

This results could be significantly improved by exploiting sparsity in at least two ways. First, the states for the lower-bounding problem could be arranged in the order

$$(x_1^L, x_1^U, x_1^{cv}, x_1^{cc}, ..., x_{n_x}^L, x_{n_x}^U, x_{n_x}^{cv}, x_{n_x}^{cc})$$

and a banded linear solver with an upper half-bandwidth of 1 and a lower half-bandwidth of 7 could be used in the numerical integration, where the number of integration variables for the lower-bounding problem is $4n_x$. (The lower and upper half-bandwidths (m_{lower} and m_{upper}) are such that every nonzero element (i, j) of the Jacobian matrix for the ODE vector field satisfies $-m_{\text{lower}} \leq j - i \leq m_{\text{upper}}$.) For the upper-bounding problem, the upper half-bandwidth would be 0 and the lower half-bandwidth would be 1. Second, in the vector field evaluation, when applying the $\mathcal{B}_i^{L/U}$ operators [172, Definition 2] to the current values of the bounds and the $\mathcal{R}_i^{cv/cc}$ operators [174, Definition 11] to the current values of the relaxations, the current implementation copies the entire vector of bounds or relaxations, then modifies the appropriate component. The time required for this scales as n_x. A more

Table 2.18: Pseudorandom noise added to state data to create a parameter estimation problem

t	$\widehat{x}_{n_x-1}(t) - x_{n_x-1}(t,p)$	$\widehat{x}_{n_x}(t) - x_{n_x}(t,p)$
0.1	-0.0117	-0.0532
0.2	-0.0339	-0.0931
0.3	-0.0448	-0.0579
0.4	0.1009	0.0133
0.5	0.0661	0.0156
0.6	0.0123	0.0180
0.7	-0.0502	-0.0052
0.8	-0.0346	0.0073
0.9	-0.0071	-0.0190
1.0	0.0316	0.0344

efficient, but more error-prone, method is as follows. Modify the vector without copying it, keeping temporary variables for the index i, the direction $(L/U/cv/cc)$, and the value of the appropriate ith value before applying the operator, so that the effects of the $\mathcal{B}_i^{L/U}$ and $\mathcal{R}_i^{cv/cc}$ operators can be reversed. The time required for this method does not depend on n_x.

2.3 Discussion and conclusions

Our software dGDOpt solves chemical kinetics problems at least as quickly as all other deterministic global dynamic optimization methods in the literature [114, 162, 185, 187] and in some cases 10 to 50 times faster than [162]. For optimal control problems, our software tends to scale better with the number of control parameters than methods based on Taylor models [116, 162] but the methods based on Taylor models tend to solve problems with smaller numbers of control parameters more quickly than our methods.

We explored some possibilities for further improvements to the methods. For the singular control problem with $n_p = 5$, using the GradBV rule for selecting the branch variable rather than selecting the variable with the largest absolute diameter gave 29% lower node counts and commensurate improvement in CPU times. Also for the singular control problem, decreasing the frequency of the upper-bounding problem solutions from every level of the B&B tree to every third level and using NaïveBds instead of the usual bounds decreased

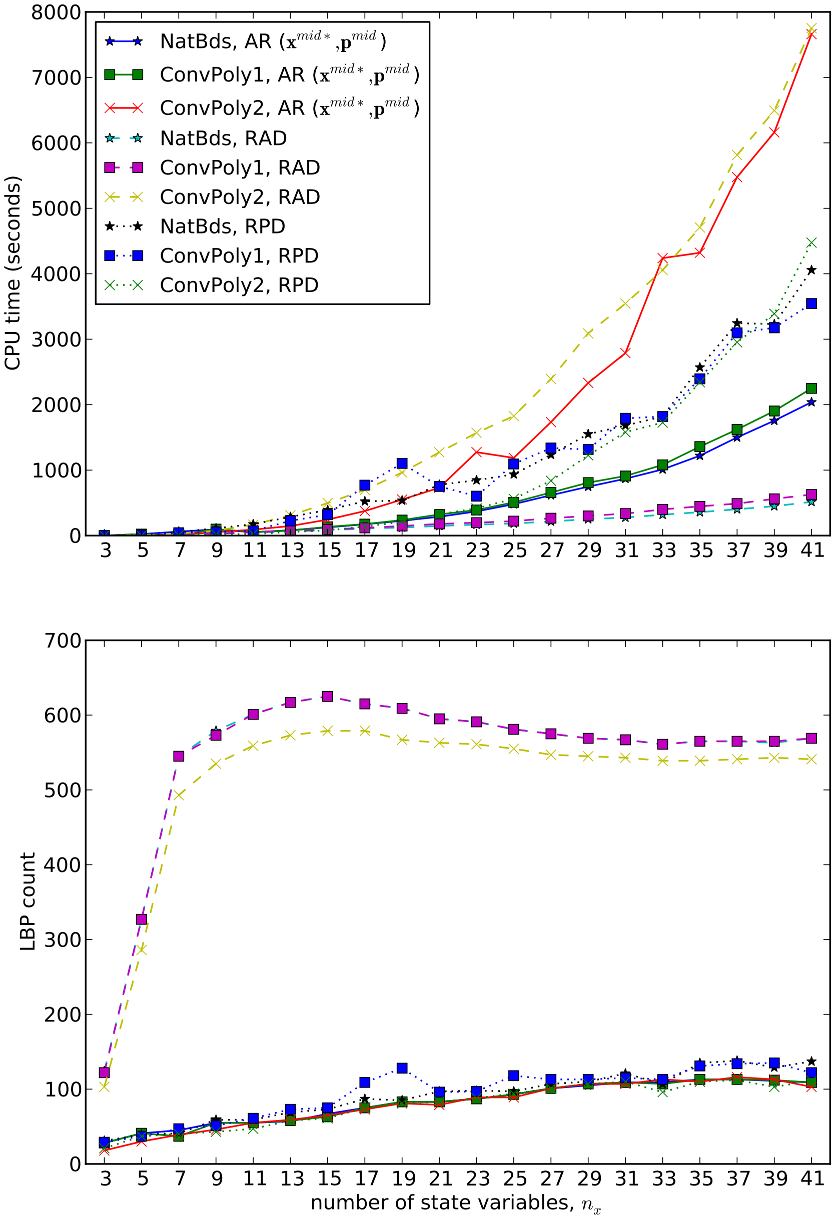


Figure 2-5: Scaling of CPU time and LBP count with n_x for discretized PDE example (§2.2.7) solved using CPLEX for the LBP

the CPU time required by a factor of 1.06–1.86 for n_p ranging from 1 to 5. This suggests that the speed of dGDOpt could be further improved by only performing the flattening operations in RPD and the state bounding systems when it is beneficial, which is exactly for those states i such that $\dot{x}_i$ depends on x_i. For a few problems, we experimented with the linear programming (LP) solver CPLEX for the lower-bounding problems rather than the nonlinear programming (NLP) solver SNOPT. For the Denbigh problem with $n_p = 3$ for NatBds and RPD, solving as a LP with CPLEX gave a 23-fold reduction in CPU time compared to solving as a NLP with SNOPT. For the fed-batch control problem with $n_p = 7$, we observed up to a 5-fold reduction from using the LP solver rather than the NLP solver.

Concerning the local optimizer used to solve the lower-bounding problems, our software sometimes gave faster results when formulating the lower-bounding problem as a LP and solving with CPLEX rather than treating it as a NLP and solving with SNOPT. In addition, the lower-bounding problems are in general nonsmooth, so it is more rigorous to solve the lower-bounding problem by linearizing and using CPLEX since SNOPT is designed for problems in which the objective function and constraints are twice continuously differentiable and neither the LP nor NLP formulations meet this qualification in general.

Among the auxiliary-ODE-based relaxation methods, the RPD relaxation method is typically the best choice because it either gives a similar CPU time to the fastest method, or a much faster CPU time—up to 10× faster than the slowest method for a given test problem, holding all other optimization options fixed. In future implementations, we suggest using RPD and only performing the flattening operations where beneficial, as described earlier. We think this would make the CPU time of RPD the fastest among AR, RAD, and RPD in almost all problems.

GradBV is almost always a better heuristic to select the branching variable than AbsDiamBV. When GradBV does not give the fastest results, they are typically within a few percentage points of the results with AbsDiamBV. On the other hand, the absolute diameter branch variable selection heuristic can be 200 times slower.

It is difficult to choose which bounding method among NatBds, ConvPoly1, and ConvPoly2 is best, because each of them gave the fastest performance for at least one problem. When considering only RPD relaxations, those respective problems are oil shale pyrolysis, the reformulated fed-batch control problem for $n_p = 1, 3, 5$, and the Denbigh problem with $n_p = 2, 3$. However, for chemical kinetics problems ConvPoly2 tends to yield the fastest CPU times in our experience.

There remain several interesting areas for exploration in global dynamic optimization, especially extending to dynamic systems the remaining ideas from global optimization of algebraic systems. Heuristics for priority-ranking the next nodes to process, variables to branch on and perform domain reduction on, and locations for branching have been carefully developed for branch-and-bound and its relatives [21, 194, 195] and can strongly impact the structure of the branch-and-bound tree as well as overall CPU time requirements. Along the same lines as the GradBV heuristic developed here, we suggest extending more of these ideas to problems with dynamics embedded as a critical step for further improving global dynamic optimization. Another interesting avenue for future research, specific to optimal control problems, is to apply branch-and-lift [85] using dGDOpt for the bounds and relaxations on the underlying ODEs. The basic idea of branch-and-lift is to solve an optimal control problem using orthogonal basis functions, first parameterizing the controls by one parameter and finding some set known to contain the global optimum for that parameter, then adding the second control parameter and optimizing again with the first parameter already known to lie in some reduced domain, and continuing to add parameters in this way, reducing the domain before adding each additional control parameter.

Another possible avenue for improving the CPU efficiency of global dynamic optimization is to dynamically switch between different possible bounding and relaxation methods. The quality of both the bounds and the relaxations impact the final relaxation of the objective function. A problem can be considered sufficiently hard to solve if the number of nodes remaining in the B&B tree exceeds some threshold. To choose which bounding and relaxation method to use, an empirical convergence order analysis can be performed near the best-known solution to the upper-bounding problem. If the more expensive bounding (ConvPoly) or relaxation (RPD) methods give a significant improvement over the methods which are cheaper per node for a node of the current diameter, then they would be enabled.

2.4 Acknowledgments

We thank Novartis Pharmaceuticals for financial support. The discretized PDE problem with a single optimization decision and a variable number of states was inspired by personal communication with Alexander Mitsos (RWTH Aachen University).

2.5 Availability of software

The software dGDOpt is available from the authors upon request.

Chapter 3

Convergence analysis for differential-inequalities-based bounds and relaxations of the solutions of ODEs

Abstract

For the tractability of global optimization algorithms, the rate of convergence of convex relaxations to the objective and constraint functions is critical. We extend results from Bompadre and Mitsos (*J. Glob. Optim.* 52(1): 1–28, 2012) to characterize the convergence rate of parametric bounds and relaxations of the solutions of ordinary differential equations (ODEs). Such bounds and relaxations are used for global dynamic optimization and are computed using auxiliary ODE systems that use interval arithmetic and McCormick relaxations. Two ODE bounding methods are shown to give first-order convergence. Two ODE relaxation methods (Scott, Chachuat, and Barton (*Optim. Control Appl. and Meth.* 34(2), 145–163, 2013); Scott and Barton (*J. Glob. Optim.* 57:143–176, 2013)) are shown to give second-order convergence, yet they can behave very differently from each other in practice. As time progresses, the prefactor in the convergence-order bound tends to grow much more slowly for one of the methods, and can even decrease over time, yielding global optimization procedures that require significantly less CPU time.

3.1 Introduction

Dynamic optimization or *optimal control* problems seek to minimize an objective function that depends on the solution of an ordinary differential equation (ODE) or differential-algebraic equation (DAE) system. There are numerous applications such as minimizing the cost of a chemical process containing a plug-flow reactor or batch reactor; optimizing the steering, braking, and acceleration of a self-driving car to conserve fuel or minimize time to a destination while obeying safety constraints; finding the worst-case behavior of one of these systems for safety analysis; or identifying the best-fit parameters to a dynamic model. Many methods have been developed for local dynamic optimization, but for some applications such as safety analysis or rejecting a model that fails to accurately predict system behavior [188], a certificate of global optimality is essential. For some dynamic optimization problems, such as those involving chemical reactions, local optima can be very numerous [121, 156], rendering local optimization techniques ineffective.

This chapter provides an analysis of the convergence order of methods for solving dynamic optimization problems to global optimality. We address dynamic models containing ODEs only. Local dynamic optimization with DAEs embedded is also possible [46, 110, 206], as is rigorous deterministic global optimization with DAEs embedded [170, 173, 175, 176], but both are outside the scope of this thesis.

Global optimization techniques can be broadly divided into stochastic and deterministic methods. Several stochastic global dynamic optimization techniques have been proposed, such as [14, 131, 156], but they cannot rigorously guarantee global optimality and we do not consider them further. Deterministic global dynamic optimization techniques rely on methods of generating pointwise-in-time lower and upper bounds on the parametric solution of the ODEs on successively refined subsets of the decision space. Some techniques also generate convex relaxations to the parametric solution. For non-dynamic global optimization problems it has been observed that convex relaxations, rather than interval bounds alone, can significantly improve the performance of global optimization methods due to their higher convergence order and the enhanced ability to apply domain reduction techniques. The present contribution shows that these advantages extend to the dynamic optimization methods studied.

Two major approaches have been proposed for generating the necessary bounds and relaxations for deterministic global dynamic optimization. The first is based on Taylor models [86, 114, 116, 117, 162–164]. A convergence analysis for bounds and relaxations generated by these methods has been recently published [33]. The second major approach uses auxiliary ODE systems that, when solved, provide the required bounds and relaxations [48, 171, 172, 174, 177, 186, 187]. The auxiliary-ODE approach uses interval arithmetic [134–136] and generalized McCormick relaxations [125, 178] to obtain global information on the variation of functions of interest on a desired domain with low computational cost. In the present contribution, we develop convergence-order bounds for this auxiliary ODE approach.

Convergence order in various senses is often a key measure of the performance of numerical algorithms. In this thesis, by *convergence order* we mean the rate at which bounds or relaxations of a function (such as a state variable, objective function, or constraint function) approach the (possibly nonconvex) function being relaxed as the size of the domain is reduced (Definitions 3.2.17 and 3.2.19). Convergence order in this sense has long been studied for interval analysis [4, 134, 135, 167] and has very recently been formalized for McCormick-based [125, 178] convex relaxations [32].

The number of nodes in a branch-and-bound routine depends strongly on the convergence order [62] as well as the convergence order prefactor [214]. Several estimates for the scaling relationship have been published. Schöbel et al. [166, Theorem 2] present an upper bound on the number of boxes considered in a branch-and-bound routine as a sum of the boxes considered in each level of the tree, assuming that the search space is divided into 2^n congruent boxes. They assume that the convergence order bound (Definition 3.2.17) for the relaxations for the objective function holds with equality. They make very weak hypotheses, neglecting even the curvature of the objective function, arriving at an upper bound that is broadly applicable but potentially very conservative. At the other limit, we can assume that boxes cluster in the vicinity of the global minimum, and that those boxes dominate the total number of boxes, as in Du and Kearfott [62]. This “clustering” analysis can be used to show that first-order-converging bounding methods give exponential scaling with the dimension

Table 3.1: Scaling of number of boxes in a branch-and-bound routine (with branch-and-bound absolute tolerance ε_{BB}) differs depending on which regime dominates. For most problems, the true scaling tends to behave between these limiting cases. n is problem dimension and order refers to the order of Hausdorff convergence in P of the bounding method.

Order	Boxes Uniformly Distributed [166]	Boxes Near Minima Dominate [214]
1	$\propto (\varepsilon_{\mathrm{BB}}^{-n})$	$\propto (\varepsilon_{\mathrm{BB}}^{-\frac{n}{2}})$
2	$\propto (\varepsilon_{\mathrm{BB}}^{-n/2})$	$\propto (1)$
3	$\propto (\varepsilon_{\mathrm{BB}}^{-n/3})$	$\propto (\varepsilon_{\mathrm{BB}}^{\frac{n}{6}})$

of the optimization problem whereas second-order-converging bounding methods give weak scaling with problem dimension and third-order-converging bounding methods give inverse scaling of the number of boxes in the clustering region with problem dimension. It is widely believed that generating third-order-converging methods is NP-hard [142], so second-order methods are the most attractive in practice. Some illustrative cases for the two analyses are presented in Table 3.1. Refining the analysis of [62], Wechsung et al [214, Theorem 1] showed that using branch-and-bound with a bounding method with convergence order of 2, the exponential scaling of the number of boxes with problem dimension can be prevented by a sufficiently small convergence order prefactor.

In this chapter, we analyze the convergence order of the differential-inequalities-based ODE bounding method developed in [81, 172, 186] as well as the ODE relaxation method developed in [177] and subsequently improved in [174]. The developments are organized as follows. In Section 3.2, we give some necessary definitions and lemmas, including two notions of convergence order. In Section 3.3, we state the problem of finding convergence-order bounds on the bounds and relaxations of the solutions of ODEs. In Section 3.4, we show that the methods from [81, 172, 186] produce bounds on the solutions of parametric ODEs that converge at least linearly as the parameter space is refined (Theorem 3.4.10). Next we use the logarithmic norm [59, 190] to show that, under appropriate conditions, the bounds can actually become tighter as time increases—the convergence-order prefactor can decrease over time and asymptotically approach a fixed value as time tends to infinity (Theorem 3.4.15). In Section 3.5, we show that the methods described in [174, 177] produce relaxations of the parametric solutions of ODEs that converge quadratically as the parameter space is refined (Theorem 3.5.9). We next use the logarithmic norm to show that for the improved method [174], the convergence-order prefactor can decrease as time progresses and asymptotically approach a fixed value (Theorem 3.5.17). Relaxations constructed using the improved relaxation method [174] have much more favorable dependence on time than relaxations constructed using the older relaxation method [177]. For the improved method, the prefactor can decrease over time, whereas for the older method, the prefactor can never decrease once a certain level of conservatism has been introduced. In Section 3.6, we discuss a numerical example. Concluding remarks are given in Section 3.7.

3.2 Preliminaries

In this section, some necessary definitions and previously-published lemmas and theorems are given. Much of the content of this section comes from [32, 33] with minor extensions to vector-valued functions. These preliminaries will be applied and extended in the subsequent sections.

3.2.1 Basic notation and standard analysis concepts

For notation, we use lower-case italic letters for scalars and scalar-valued functions. We use lower-case bold letters for vectors and vector-valued functions. We use capital letters for sets and set-valued functions.

Definition 3.2.1 (Lipschitz and locally Lipschitz functions)**.** Let (X, d_X) and (Z, d_Z) be metric spaces. Let $\widehat{X} \subset X$. A function $f : X \to Z$ is said to be *Lipschitz on* $\widehat{X}$ *with Lipschitz constant* M if

$$d_Z(f(x_1), f(x_2)) \leq M d_X(x_1, x_2), \quad \forall x_1, x_2 \in \widehat{X},$$

and $M \in \mathbb{R}_+$ is the smallest value for which the inequality holds. f is *locally Lipschitz* if, for every $\widehat{x} \in X$, $\exists \eta, M \in \mathbb{R}_+$ such that

$$d_Z(f(x_1), f(x_2)) \leq M d_X(x_1, x_2), \quad \forall x_1, x_2 \in B_\eta(\widehat{x}),$$

where $B_\eta(\widehat{x}) \equiv \{x \in X : d_X(x, \widehat{x}) < \eta\}$ is the open ball in X of radius η about $\widehat{x}$. Let $I \subset \mathbb{R}$ and $g : I \times X \to Z$. The function g is said to be *Lipschitz on $\widehat{X}$, uniformly on I* if $\exists M \in \mathbb{R}_+$ such that

$$d_Z(g(t, x_1), g(t, x_2)) \leq M d_X(x_1, x_2), \quad \forall (t, x_1, x_2) \in I \times \widehat{X} \times \widehat{X},$$

and *locally Lipschitz on X, uniformly on I if*

$$d_Z(g(t, x_1), g(t, x_2)) \leq M d_X(x_1, x_2), \quad \forall (t, x_1, x_2) \in I \times B_\eta(\widehat{x}) \times B_\eta(\widehat{x}).$$

Proposition 3.2.2. *Let (X, d_X) and (Y, d_Y) be metric spaces and let $f : X \to Y$. If f is locally Lipschitz on X then f is Lipschitz on every compact $K \subset X$.*

Lemma 3.2.3 (Gronwall-Bellman inequality [94, Lemma A.1])**.** *If $x : I \to \mathbb{R}$ and $\lambda : I \to \mathbb{R}$, and $\mu : I \to \mathbb{R}$ are continuous functions, μ is nonnegative, and x satisfies*

$$x(t) \leq \lambda(t) + \int_{t_0}^{t} \mu(s) x(s) \mathrm{d}s, \quad \forall t \in I,$$

then

$$x(t) \leq \lambda(t) + \int_{t_0}^{t} \lambda(s) \mu(s) \exp\left[\int_{s}^{t} \mu(\tau) d\tau\right] \mathrm{d}s, \quad \forall t \in I.$$

If $\lambda \equiv \lambda(t)$ is a constant, then

$$x(t) \leq \lambda \exp\left[\int_{t_0}^{t} \mu(\tau) d\tau\right].$$

If, in addition, $\mu \equiv \mu(t) \geq 0$ is a constant, then

$$x(t) \leq \lambda \exp[\mu(t - t_0)].$$

The following result is similar to Lemma 3.2.3 but with μ allowed to be a negative constant. We will use it to show that the diameter of the state bounds can become smaller with increasing time.

Lemma 3.2.4. *Let $\lambda_0, \lambda_1 \in \mathbb{R}_+$, $\mu \in \mathbb{R}$. Let $x : I \to \mathbb{R}$ be continuous. If x satisfies*

$$x(t) \leq \lambda_0 + \lambda_1(t - t_0) + \int_{t_0}^{t} \mu x(s)\mathrm{d}s, \quad \forall t \in I, \tag{3.1}$$

then

$$\begin{cases} x(t) \leq \left(\lambda_0 + \frac{\lambda_1}{\mu}\right) \exp(\mu(t - t_0)) - \frac{\lambda_1}{\mu} & \text{if } \mu \neq 0, \\ x(t) \leq \lambda_0 + \lambda_1(t - t_0) & \text{if } \mu = 0, \end{cases}$$

for all $t \in I$.

Proof. See §3.9. □

The logarithmic norm [59, 190] is useful for producing tighter bounds on convergence behavior of bounds and relaxations of parametric ODEs than would be possible without it. Note, however, that the logarithmic norm of a matrix is not truly a norm since it can be negative.

Definition 3.2.5 (Logarithmic norm of a matrix)**.** Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $p \in \{1, 2, \infty\}$. The *logarithmic norm* of $\mathbf{A}$ is given by

$$\mu_p(\mathbf{A}) \equiv \lim_{h \to 0^+} \frac{\|\mathbf{I} + h\mathbf{A}\|_p - 1}{h},$$

where $\|\mathbf{A}\|_p$ is the induced p-norm of $\mathbf{A}$.

Proposition 3.2.6. *The following formulas are equivalent to the definition of the logarithmic norm:*

$$\begin{aligned} \mu_1(\mathbf{A}) &= \max_{i=1,\dots,n} \left(a_{ii} + \sum_{k \neq i} |a_{ki}| \right), \\ \mu_2(\mathbf{A}) &= \lambda_{max} = \textit{largest eigenvalue of } \frac{1}{2}(\mathbf{A}^{\mathrm{T}} + \mathbf{A}), \\ \mu_\infty(\mathbf{A}) &= \max_{i=1,\dots,n} \left(a_{ii} + \sum_{k \neq i} |a_{ik}| \right). \end{aligned}$$

Proof. See [59, §1.2] or [80, Theorem I.10.5]. □

For μ_1 and μ_∞, these are equivalent to their regular norm counterparts, except that the absolute values of the diagonal terms are not taken.

3.2.2 Interval analysis

Definition 3.2.7 (Interval)**.** For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, $\mathbf{a} \leq \mathbf{b}$, define the interval $[\mathbf{a}, \mathbf{b}]$ as the compact, connected set $\{\mathbf{p} \in \mathbb{R}^n : \mathbf{a} \leq \mathbf{p} \leq \mathbf{b}\}$. It can be written $P = [\mathbf{p}^L, \mathbf{p}^U] = [p_1^L, p_1^U] \times \cdots \times [p_n^L, p_n^U]$. We will sometimes use a single real vector to denote a singleton interval, as in $\mathbf{p} \equiv [\mathbf{p}, \mathbf{p}]$. The set of all interval subsets of $D \subset \mathbb{R}^n$ is denoted $\mathbb{I}D$. In particular, $\mathbb{IR}^n$ is the set of all interval subsets of $\mathbb{R}^n$.

Definition 3.2.8 (Diameter of a set)**.** The *diameter* of a set $P \subset \mathbb{R}^n$ is given by:

$$w(P) = \sup_{\mathbf{p}_1, \mathbf{p}_2 \in P} \|\mathbf{p}_1 - \mathbf{p}_2\|_\infty.$$

Occasionally, we will want to know the diameter of each component of a set. We denote by $w_V(P)$ a vector of diameters for each component. Given $P \subset \mathbb{R}^n$, $w_V(P) \in \mathbb{R}^n$, where the ith component is given by

$$\sup_{\mathbf{p}_1, \mathbf{p}_2 \in P} |p_{1,i} - p_{2,i}|.$$

Definition 3.2.9 (Hausdorff metric)**.** For $X, Y \subset \mathbb{R}^n$ compact and nonempty, the Hausdorff metric is given by

$$d_H(X, Y) \equiv \max \left\{ \sup_{\mathbf{x} \in X} \inf_{\mathbf{y} \in Y} \|\mathbf{x} - \mathbf{y}\|_\infty, \sup_{\mathbf{y} \in Y} \inf_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{y}\|_\infty \right\}.$$

When the argument sets are both intervals, the Hausdorff metric specializes as follows:

Proposition 3.2.10 (Hausdorff metric for intervals)**.** *Let $X = [\mathbf{x}^L, \mathbf{x}^U]$ and $Y = [\mathbf{y}^L, \mathbf{y}^U]$ be two intervals in $\mathbb{IR}^n$. Then the Hausdorff metric, $d_H(X, Y)$ is equivalent to:*

$$d_H(X, Y) = \max_{i \in \{1, \ldots, n\}} \max \left\{ |x_i^L - y_i^L|, |x_i^U - y_i^U| \right\} = \max_{i \in \{1, \ldots, n_x\}} d_H(X_i, Y_i).$$

Definition 3.2.11 (Image and inclusion function)**.** Let $P \subset \mathbb{R}^{n_p}$ be nonempty. Consider a vector-valued continuous function $\mathbf{f} : P \to X \subset \mathbb{R}^{n_x}$. The *image* of $\widehat{P} \subset P$ under $\mathbf{f}$ is denoted by $\mathbf{f}(\widehat{P})$. Consider also an n_x-dimensional interval-valued function $F : \mathbb{I}P \to \mathbb{IR}^{n_x}$. F is an *inclusion function* for $\mathbf{f}$ on $\mathbb{I}P$ if

$$\mathbf{f}(\widehat{P}) \subset F(\widehat{P}),\ \forall \widehat{P} \in \mathbb{I}P.$$

Definition 3.2.12 (Interval hull)**.** For any nonempty, bounded set $P \subset \mathbb{R}^n$, let the *interval hull* of P be denoted by $\square P$. This is the smallest interval containing P; it can be written

$$\square P \equiv [\inf\{p_1 : \mathbf{p} \in P\}, \sup\{p_1 : \mathbf{p} \in P\}] \times \cdots \times [\inf\{p_n : \mathbf{p} \in P\}, \sup\{p_n : \mathbf{p} \in P\}].$$

3.2.3 Convex relaxations

Definition 3.2.13. Let $\mathbf{f} : P \to \mathbb{R}^{n_x}$ be a continuous function. A function $\mathbf{u} : P \to \mathbb{R}^{n_x}$ is said to be a *convex relaxation* (or equivalently a *convex underestimator*) of $\mathbf{f}$ if $\mathbf{u}(\mathbf{p}) \leq \mathbf{f}(\mathbf{p})$, $\forall \mathbf{p} \in P$ and $\mathbf{u}$ is convex. A function $\mathbf{o} : P \to \mathbb{R}^{n_x}$ is said to be a *concave relaxation* (or equivalently a *concave overestimator*) of $\mathbf{f}$ if $\mathbf{o}(\mathbf{p}) \geq \mathbf{f}(\mathbf{p})$, $\forall \mathbf{p} \in P$ and $\mathbf{u}$ is concave.

McCormick [125] defined a method for computing convex and concave relaxations for a large class of functions—any function that can be decomposed into a finite sequence of addition, multiplication, and univariate composition operations. This relaxation method has been further formalized in [170, Chapter 2]. We next define $\mathbb{MR}^n$, which is the space containing the basic mathematical objects in McCormick's relaxation technique [125] as formalized in [170, Chapter 2].

Definition 3.2.14 ([170])**.** Let $D \subset \mathbb{R}^n$. A set $\mathbb{M}D$ is denoted

$$\mathbb{M}D \equiv \{(Z^B, Z^C) \in \mathbb{I}D \times \mathbb{I}D : Z^B \cap Z^C \neq \emptyset\}.$$

Elements of $\mathbb{MR}^n$ are denoted by script capitals $\mathcal{Z}$. They will also be referred to using Z^B and Z^C. To save space, for a "thin" McCormick object $([\mathbf{a}, \mathbf{a}], [\mathbf{a}, \mathbf{a}])$, we will sometimes simply use $\mathbf{a}$.

Definition 3.2.15 (Relaxation function)**.** Let $P \subset \mathbb{R}^{n_p}$ be nonempty and let $\mathbf{f} : P \to \mathbb{R}^{n_x}$ be a continuous function. Suppose we can construct two functions $\mathbf{f}^{cv}, \mathbf{f}^{cc} : \mathbb{M}P \to \mathbb{R}^{n_x}$ such that for each $(i, \widehat{P}) \in \{1, \dots, n_x\} \times \mathbb{I}P$:

- the function $u_i : \mathbf{p} \mapsto f_i^{cv}((\widehat{P}, [\mathbf{p}, \mathbf{p}]))$ is a convex underestimator of f_i on $\widehat{P}$,
- the function $o_i : \mathbf{p} \mapsto f_i^{cc}((\widehat{P}, [\mathbf{p}, \mathbf{p}]))$ is a concave overestimator of f_i on $\widehat{P}$.

We call the pair of functions $(\mathbf{f}^{cv}, \mathbf{f}^{cc})$ a *relaxation function* for $\mathbf{f}$ in P. We call the relaxation function *continuous* if $f_i^{cv}((\widehat{P}, \cdot)), f_i^{cc}((\widehat{P}, \cdot))$ are continuous for all $(\widehat{P}, i)$. To streamline notation, we also define $F^C \equiv [\mathbf{f}^{cv}, \mathbf{f}^{cc}]$ (using the corresponding capital letter).

Bompadre and Mitsos [32] use the term *scheme of estimators* instead of *relaxation function.*

Throughout the rest of the chapter, the notation $v^{cv/cc}$ is used to indicate that any associated statement holds independently for both v^{cv} and v^{cc}.

Definition 3.2.16 (Inclusion function associated to a relaxation function)**.** Let $P \subset \mathbb{R}^{n_p}$ be nonempty and let $\mathbf{f} : P \to \mathbb{R}^{n_x}$ be a continuous function. Let $F^C : \mathbb{M}P \to \mathbb{IR}^{n_x}$ be a relaxation function for $\mathbf{f}$ in P. The inclusion function associated to this relaxation function is:

$$\begin{aligned} H_{\mathbf{f}} : \mathbb{I}P \to \mathbb{IR}^{n_x} : \widehat{P} \mapsto & \left[\inf_{\mathbf{p} \in \widehat{P}} f_1^{cv}((\widehat{P}, [\mathbf{p}, \mathbf{p}])), \sup_{\mathbf{p} \in \widehat{P}} f_1^{cc}((\widehat{P}, [\mathbf{p}, \mathbf{p}]))\right] \times \cdots \\ & \times \left[\inf_{\mathbf{p} \in \widehat{P}} f_{n_x}^{cv}((\widehat{P}, [\mathbf{p}, \mathbf{p}])), \sup_{\mathbf{p} \in \widehat{P}} f_{n_x}^{cc}((\widehat{P}, [\mathbf{p}, \mathbf{p}]))\right]. \end{aligned}$$

3.2.4 Convergence order

Definition 3.2.17 (Hausdorff convergence order and prefactor)**.** Let $P \subset \mathbb{R}^{n_p}$ be nonempty. Let $\mathbf{f} : P \to \mathbb{R}^{n_x}$ be a continuous function, and let F be an inclusion function for $\mathbf{f}$ on $\mathbb{I}P$. The inclusion function F has *Hausdorff convergence in* P *of order* β *with prefactor* τ if there exist constants $\tau, \beta > 0$ such that

$$d_H\left(\Box\mathbf{f}(\widehat{P}), F(\widehat{P})\right) \leq \tau w\left(\widehat{P}\right)^{\beta}, \quad \forall \widehat{P} \in \mathbb{I}P. \tag{3.2}$$

Let $I \subset \mathbb{R}$ and $\mathbf{g} : I \times P \to \mathbb{R}^{n_x}$. Let $G(t, \cdot)$ be an inclusion function for $\mathbf{g}(t, \cdot)$ on $\mathbb{I}P$ for every $t \in I$. If

$$d_H\left(\Box\mathbf{g}(t, \widehat{P}), G(t, \widehat{P})\right) \leq \tau w\left(\widehat{P}\right)^{\beta}, \quad \forall (t, \widehat{P}) \in I \times \mathbb{I}P,$$

then G is said to have Hausdorff convergence in P of order β with prefactor τ *uniformly on* I.

Definition 3.2.18 ((γ_1, γ_2)-convergence)**.** For $\mathcal{X} \in \mathbb{MR}^n$, let $w(\mathcal{X}) \equiv w(\text{Enc}(\mathcal{X})) \equiv w(X^B \cap X^C)$. Let $\mathcal{F} : \mathbb{M}X^0 \subset \mathbb{MR}^n \to \mathbb{MR}^m$. We say that $\mathcal{F}$ has (γ_1, γ_2)-convergence on $\mathbb{M}X^0$ if $\exists \tau_1, \tau_2 \in \mathbb{R}_+$ such that

$$w(\mathcal{F}(\mathcal{X})) \leq \tau_1 w(\mathcal{X})^{\gamma_1} + \tau_2 w(X^B)^{\gamma_2}, \quad \forall \mathcal{X} \in \mathbb{M}X^0. \tag{3.3}$$

In §3.9.7, we show that natural McCormick extensions [125, 170, 178] have $(1, 2)$-convergence.

Definition 3.2.19 (Pointwise convergence order and prefactor)**.** Let $P \subset \mathbb{R}^{n_p}$ be nonempty and $\mathbf{f} : P \to \mathbb{R}^{n_x}$ be continuous. Let $F^C : \mathbb{M}P \to \mathbb{IR}^{n_x}$ be a relaxation function for $\mathbf{f}$ in P. The relaxation function has *pointwise convergence in* P *of order* γ *with prefactor* τ if there exist constants $\tau, \gamma > 0$ such that

$$\sup_{\mathbf{p} \in \widehat{P}} w\left(F^C((\widehat{P}, [\mathbf{p}, \mathbf{p}]))\right) \leq \tau w\left(\widehat{P}\right)^{\gamma}, \quad \forall \widehat{P} \in \mathbb{I}P.$$

Let $I \subset \mathbb{R}$ and $\mathbf{g} : I \times P \to \mathbb{R}^{n_x}$. Let $G^C(t, (\widehat{P}, \cdot))$ be a relaxation function for $\mathbf{g}(t, \cdot)$ in P for every $t \in I$. If

$$\sup_{\mathbf{p} \in \widehat{P}} w\left(G^C(t, (\widehat{P}, [\mathbf{p}, \mathbf{p}]))\right) \leq \tau w\left(\widehat{P}\right)^{\gamma}, \quad \forall (t, \widehat{P}) \in I \times \mathbb{I}P,$$

then G^C has pointwise convergence in P of order γ with prefactor τ, *uniformly on* I.

The reader can verify that pointwise convergence in the sense of Definition 3.2.19 and pointwise convergence in the sense of [32] are equivalent to within a factor of two. Further-

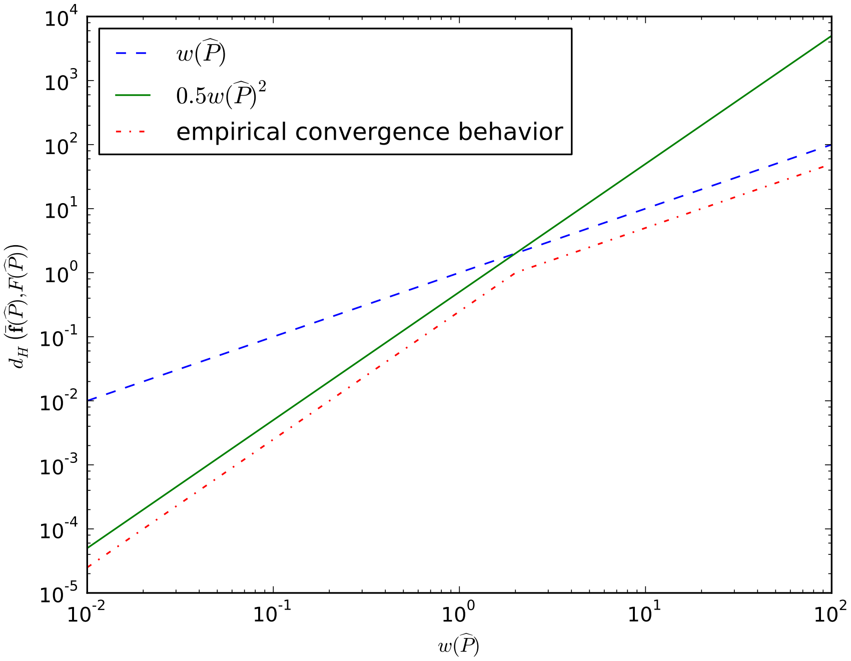


Figure 3-1: This hypothetical empirical convergence behavior satisfies a linear convergence bound yet still also satisfies a quadratic convergence bound for any $\widehat{P} \in \mathbb{I}P$. In this way, convergence of a given order does not preclude convergence of higher order. On larger sets the linear bound is stronger than the quadratic bound; on smaller sets, the quadratic bound is stronger.

more, pointwise convergence is a special case of (γ_1, γ_2)-convergence (to within a factor of two) with X^C degenerate.

In the definitions of Hausdorff and pointwise convergence orders and prefactors, the order and prefactor may depend on the host set P but not on the intervals $\widehat{P}$. Also, the definitions allow for the possibility that the convergence orders are not the highest possible. Whereas a convergence order bound can be very weak, we will use the term "empirical convergence behavior" to indicate the curve along which the Hausdorff distance $d_H(\Box\mathbf{f}(\widehat{P}), F(\widehat{P}))$ actually passes as the diameter of the host interval $\widehat{P}$ is decreased. See Figure 3-1.

3.3 Problem statement

We are interested in bounds and relaxations of the solution of the following ODE.

Problem 3.3.1. Let $I = [t_0, t_f] \subset \mathbb{R}$ be the time interval of interest, $D \subset \mathbb{R}^{n_x}$ be an open, connected set, $P \subset \mathbb{R}^{n_p}$ be the set of possible parameter values, $\mathbf{f} : I \times D \times P \to \mathbb{R}^{n_x}$ be the vector field for the ODE, and $\mathbf{x}_0 : P \to D$ be the initial condition. We are interested in the solution of the initial value problem (IVP) in ODEs:

$$\begin{aligned} \dot{\mathbf{x}}(t, \mathbf{p}) &= \mathbf{f}(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}), \quad \forall t \in (t_0, t_f], \\ \mathbf{x}(t_0, \mathbf{p}) &= \mathbf{x}_0(\mathbf{p}). \end{aligned} \tag{ODE}$$

For the ODE relaxation theory to be valid and to obtain the needed convergence order bounds, we assume:

Assumption 3.3.2. *Problem 3.3.1 satisfies the following conditions:*

1. *A unique solution exists on all of $I \times P$ and*

2. *$\mathbf{x}_0$ and $\mathbf{f}$ are locally Lipschitz.*

We show in §3.9.3 that if $\mathbf{x}_0$ and $\mathbf{f}$ are factorable functions and all of the univariate functions in the factored representation are locally Lipschitz, then $\mathbf{x}_0$ and $\mathbf{f}$ are locally Lipschitz as well. With $\mathbf{x}$ now well-defined, we seek bounds on the convergence order (Definitions 3.2.17 and 3.2.19) of state bounds and state relaxations of the solution of Problem 3.3.1. State bounds and relaxations are defined below.

Definition 3.3.3 (State bounds)**.** A function

$$X^B : I \times \mathbb{I}P \to \mathbb{IR}^{n_x} : (t, \widehat{P}) \mapsto [\mathbf{x}^L(t, \widehat{P}), \mathbf{x}^U(t, \widehat{P})]$$

is said to provide *state bounds* on $\widehat{P}$ if:

$$\mathbf{x}^L(t, \widehat{P}) \le \mathbf{x}(t, \mathbf{p}) \le \mathbf{x}^U(t, \widehat{P}), \quad \forall (t, \mathbf{p}) \in I \times \widehat{P},$$

where $\mathbf{x}$ is the solution of Problem 3.3.1.

Definition 3.3.4 (State relaxations)**.** A function $X^C : I \times \mathbb{M}P \to \mathbb{IR}^{n_x} : (t, (\widehat{P}, [\mathbf{p}, \mathbf{p}])) \mapsto [\mathbf{x}^{cv}(t, (\widehat{P}, [\mathbf{p}, \mathbf{p}])), \mathbf{x}^{cc}(t, (\widehat{P}, [\mathbf{p}, \mathbf{p}]))]$ is said to provide *state relaxations* if for each $t \in I$, $X^C(t, \cdot)$ is a relaxation function for $\mathbf{x}(t, \cdot)$ in P, where $\mathbf{x}$ is the solution of Problem 3.3.1.

3.4 Bounds on the convergence order of state bounds

In this section, we will study the bounds of the solutions of parametric ODEs that are computed using auxiliary ODE systems, which we refer to as *state bounding systems.* We develop convergence-order bounds on the state bounding systems. We begin by formalizing some of the necessary definitions and results for the natural interval extension.

Proposition 3.4.1. *Let $P \subset \mathbb{R}^n$. Then $(\mathbb{I}P, d_H)$ is a metric space.*

Assumption 3.4.2. *We have inclusion functions $F : \mathbb{I}I \times \mathbb{I}D \times \mathbb{I}P \to \mathbb{IR}^{n_x}$ and $X_0 : \mathbb{I}I \times \mathbb{I}P \to \mathbb{IR}^{n_x}$ for the $\mathbf{f}$ and $\mathbf{x}_0$ of Problem 3.3.1 with Hausdorff convergence of order 1 on any interval subset of their domains. Furthermore, F and X_0 are locally Lipschitz.*

The natural interval extensions of $\mathbf{f}$ and $\mathbf{x}_0$ satisfy Assumption 3.4.2, provided the technical Assumptions 3.9.4 and 3.9.6 hold [170, Theorem 2.5.30].

Next, we examine the convergence behavior of two methods for generating state bounds for Problem 3.3.1. We consider two different auxiliary systems of ODEs whose solutions provide state bounds. We consider a naïve bounding system, then a bounding system due to Harrison [81] that is based on differential inequalities [212]. Both auxiliary ODE systems can be numerically integrated to generate state bounds. We will show that Harrison's method gives bounds that can be no looser than those generated using the naïve state bounding system. Provided that Harrison's bounding method provides valid state bounds, it follows immediately that the naïve state bounding system provides valid state bounds. Next we prove that the state bounds produced by the naïve state bounding system converge linearly, and it follows that Harrison's method also produces linearly-converging state bounds.

Definition 3.4.3 (Naïve state bounding system)**.** We call the following ODE system the *naïve state bounding system* for Problem 3.3.1. For any $\widehat{P} \in \mathbb{I}P$,

$$\begin{aligned}
\dot{\mathbf{x}}^L(t,\widehat{P}) &= \mathbf{f}^L([t,t], X^B(t,\widehat{P}), \widehat{P}), \quad \forall t \in (t_0, t_f],\\
\dot{\mathbf{x}}^U(t,\widehat{P}) &= \mathbf{f}^U([t,t], X^B(t,\widehat{P}), \widehat{P}), \quad \forall t \in (t_0, t_f],\\
\mathbf{x}^L(t_0,\widehat{P}) &= \mathbf{x}_0^L(\widehat{P}),\\
\mathbf{x}^U(t_0,\widehat{P}) &= \mathbf{x}_0^U(\widehat{P}),
\end{aligned} \tag{3.4}$$

where $\mathbf{f}^{L/U}$ and $\mathbf{x}_0^{L/U}$ are lower and upper bounds from the natural interval extensions for $\mathbf{f}$ and $\mathbf{x}_0$.

Harrison's method [81] gives a computational implementation for potentially tighter bounds on the solution of Problem 3.3.1. Before we define Harrison's method, we need to define the following operator.

Definition 3.4.4 ($\mathcal{B}_i^{L/U}$)**.** Let $\mathcal{B}_i^L : \mathbb{IR}^n \to \mathbb{IR}^n : [\mathbf{v}, \mathbf{w}] \mapsto \{\mathbf{z} \in [\mathbf{v}, \mathbf{w}] : z_i = v_i\}$ and $\mathcal{B}_i^U : \mathbb{IR}^n \to \mathbb{IR}^n : [\mathbf{v}, \mathbf{w}] \mapsto \{\mathbf{z} \in [\mathbf{v}, \mathbf{w}] : z_i = w_i\}$.

Definition 3.4.5 (Harrison's method [81], as stated in [170], Equations (3.3))**.** Given an ODE as in Problem 3.3.1, Harrison's method bounds are given by the solution of the following ODE:

$$\begin{aligned}
\dot{x}_i^L(t, \widehat{P}) &= f_i^L([t,t], \mathcal{B}_i^L(X^B(t, \widehat{P})), \widehat{P}), \quad \forall t \in (t_0, t_f], \\
\dot{x}_i^U(t, \widehat{P}) &= f_i^U([t,t], \mathcal{B}_i^U(X^B(t, \widehat{P})), \widehat{P}), \quad \forall t \in (t_0, t_f], \\
[x_i^L(t_0, \widehat{P}), x_i^U(t_0, \widehat{P})] &= X_{0,i}(\widehat{P}),
\end{aligned}$$

for $i = 1, \ldots, n_x$, where $f_i^{L/U}$ are constructed using the natural interval extension of f_i and $X_{0,i}$ are constructed using the natural interval extension of $x_{0,i}$.

If a solution to the Harrison's method bounding system exists, then it provides valid state bounds [170, §3.5.3].

For the following, we denote the space of absolutely continuous functions from $[a, b]$ into $\mathbb{R}$ by $\mathcal{AC}([a, b], \mathbb{R})$. It is well-known that any $\phi \in \mathcal{AC}([a, b], \mathbb{R})$ is differentiable at almost every $t \in [a, b]$. We use the abbreviation "a.e. $t \in [a, b]$". For any measurable $I \subset \mathbb{R}$ we denote the space of Lebesgue integrable functions $u : I \to \mathbb{R}$ by $L^1(I)$.

Theorem 3.4.6. *Let $D \subset \mathbb{R}^{n_x}$ be open, $I \subset \mathbb{R}$, and $P \subset \mathbb{R}^{n_p}$. Let $\mathbf{u}, \mathbf{o}, \widetilde{\mathbf{u}}, \widetilde{\mathbf{o}} : I \times \mathbb{I}D \times \mathbb{I}P \to$*

$\mathbb{R}^{n_x}$. *Let* $\widetilde{\mathbf{u}}, \widetilde{\mathbf{o}}$ *be locally Lipschitz. Let* $\mathbf{v}_0, \mathbf{w}_0, \widetilde{\mathbf{v}}_0, \widetilde{\mathbf{w}}_0 : \mathbb{I}P \to \mathbb{R}^{n_x}$. *Suppose for all* $\widehat{P} \times \mathbb{I}P$,

$$\begin{aligned} \dot{\mathbf{v}}(t,\widehat{P}) &= \mathbf{u}(t,[\mathbf{v}(t,\widehat{P}),\mathbf{w}(t,\widehat{P})],\widehat{P}), \quad \text{a.e. } t \in (t_0,t_f] \\ \dot{\mathbf{w}}(t,\widehat{P}) &= \mathbf{o}(t,[\mathbf{v}(t,\widehat{P}),\mathbf{w}(t,\widehat{P})],\widehat{P}), \quad \text{a.e. } t \in (t_0,t_f] \\ \mathbf{v}(t_0,\widehat{P}) &= \mathbf{v}_0(\widehat{P}), \\ \mathbf{w}(t_0,\widehat{P}) &= \mathbf{w}_0(\widehat{P}), \end{aligned} \tag{3.5}$$

$$\begin{aligned} \dot{\widetilde{\mathbf{v}}}(t,\widehat{P}) &= \widetilde{\mathbf{u}}(t,[\widetilde{\mathbf{v}}(t,\widehat{P}),\widetilde{\mathbf{w}}(t,\widehat{P})],\widehat{P}), \quad \text{a.e. } t \in (t_0,t_f] \\ \dot{\widetilde{\mathbf{w}}}(t,\widehat{P}) &= \widetilde{\mathbf{o}}(t,[\widetilde{\mathbf{v}}(t,\widehat{P}),\widetilde{\mathbf{w}}(t,\widehat{P})],\widehat{P}), \quad \text{a.e. } t \in (t_0,t_f] \\ \widetilde{\mathbf{v}}(t_0,\widehat{P}) &= \widetilde{\mathbf{v}}_0(\widehat{P}), \\ \widetilde{\mathbf{w}}(t_0,\widehat{P}) &= \widetilde{\mathbf{w}}_0(\widehat{P}), \end{aligned} \tag{3.6}$$

and

$$\widetilde{\mathbf{v}}_0(\widehat{P}) \leq \mathbf{v}_0(\widehat{P}) \leq \mathbf{w}_0(\widehat{P}) \leq \widetilde{\mathbf{w}}_0(\widehat{P}). \tag{3.7}$$

Suppose furthermore

$$\begin{gathered} \widetilde{\mathbf{u}}(t,Z,\widehat{P}) \leq \mathbf{u}(t,Z,\widehat{P}) \quad \textit{and} \quad \mathbf{o}(t,Z,\widehat{P}) \leq \widetilde{\mathbf{o}}(t,Z,\widehat{P}), \\ \text{a.e. } t \in (t_0,t_f], \quad \forall (Z,\widehat{P}) \in \mathbb{I}D \times \mathbb{I}P \end{gathered} \tag{3.8}$$

and

$$\begin{gathered} \widetilde{\mathbf{u}}(t,Z,\widehat{P}) \leq \widetilde{\mathbf{u}}(t,Z',\widehat{P}) \quad \textit{and} \quad \widetilde{\mathbf{o}}(t,Z',\widehat{P}) \leq \widetilde{\mathbf{o}}(t,Z,\widehat{P}), \\ \text{a.e. } t \in (t_0,t_f], \quad \forall (Z,Z',\widehat{P}) \in \mathbb{I}D \times \mathbb{I}D \times \mathbb{I}P : Z' \subset Z. \end{gathered} \tag{3.9}$$

If solutions to (3.5) *and* (3.6) *exist on* I *then*

$$[\widetilde{\mathbf{v}}(t,\widehat{P}),\widetilde{\mathbf{w}}(t,\widehat{P})] \supset [\mathbf{v}(t,\widehat{P}),\mathbf{w}(t,\widehat{P})], \quad \forall (t,\widehat{P}) \in I \times \mathbb{I}P. \tag{3.10}$$

Proof. See §3.9.2. □

Proposition 3.4.7. *Suppose for any $\widehat{P} \in \mathbb{I}P$, solutions $\widetilde{X}^B(\cdot, \widehat{P}), X^B(\cdot, \widehat{P})$ of the naïve and Harrison state bounding systems (Definitions 3.4.3 and 3.4.5) for Problem 3.3.1 exist on I. Then the bounds from the naïve state bounding system satisfy $\widetilde{X}^B(t, \widehat{P}) \supset X^B(t, \widehat{P})$, $\forall (t, \widehat{P}) \in I \times \mathbb{I}P$.*

Proof. Fix any $\widehat{P} \in \mathbb{I}P$. Let $\widetilde{X}^B(\cdot, \widehat{P}) = [\widetilde{\mathbf{v}}(\cdot, \widehat{P}), \widetilde{\mathbf{w}}(\cdot, \widehat{P})]$ be the solution of the naïve state bounding system and $X^B(\cdot, \widehat{P}) = [\mathbf{v}(\cdot, \widehat{P}), \mathbf{w}(\cdot, \widehat{P})]$ be the solution of the Harrison's method bounding system. By Assumption 3.4.2, the vector fields for the naïve state bounding systems are locally Lipschitz. Since the initial conditions for both systems are given by the natural interval extension, (3.7) holds. Since the solutions to both systems exist, we have for each i and every $(t, Z, \widehat{P}) \in I \times \mathbb{I}D \times \mathbb{I}P$,

$$\widetilde{u}_i(t, Z, \widehat{P}) = f_i^L([t, t], Z, \widehat{P}),$$
$$u_i(t, Z, \widehat{P}) = f_i^L([t, t], \mathcal{B}_i^L(Z), \widehat{P}),$$
$$\mathcal{B}_i^L(Z) \subset Z.$$

Using the above with the fact that the natural interval extension is inclusion monotonic (and analogous facts for $\widetilde{o}_i, o_i$, and $\mathcal{B}_i^U$), it is clear that (3.8) holds. By inclusion monotonicity of the natural interval extension, (3.9) holds. Therefore, by Theorem 3.4.6, $X^B(t, \widehat{P}) \subset \widetilde{X}^B(t, \widehat{P})$, $\forall t \in I$. □

Corollary 3.4.8. *Suppose for any $\widehat{P} \in \mathbb{I}P$, solutions $\widetilde{X}^B(\cdot, \widehat{P}), X^B(\cdot, \widehat{P})$ of the naïve and Harrison state bounding systems (Definitions 3.4.3 and 3.4.5) for Problem 3.3.1 exist on I. Then the naïve state bounding method provides valid state bounds.*

Proof. This follows directly from Proposition 3.4.7. □

Theorem 3.4.9. *Consider naïve state bounds for Problem 3.3.1. If $P' \subset P$ is compact, then they have Hausdorff convergence in P' of order at least 1, uniformly on I.*

Proof. Fix any compact $P' \subset P$. Fix any $i \in \{1, \ldots, n_x\}$, $t \in I$, $\widehat{P} \in \mathbb{I}P'$, and $\mathbf{p} \in \widehat{P}$. Write the integral form of the ODE for $x_i - x_i^L$:

$$x_i(t, \mathbf{p}) - x_i^L(t, \widehat{P}) = x_{0,i}(\mathbf{p}) - x_{0,i}^L(\widehat{P}) + \int_{t_0}^{t} f_i(s, \mathbf{x}(s, \mathbf{p}), \mathbf{p}) - f_i^L(s, X^B(s, \widehat{P}), \widehat{P}) \mathrm{d}s.$$

Use the triangle inequality:

$$|x_i(t, \mathbf{p}) - x_i^L(t, \widehat{P})| \leq |x_{0,i}(\mathbf{p}) - x_{0,i}^L(\widehat{P})| + \int_{t_0}^{t} \left| f_i(s, \mathbf{x}(s, \mathbf{p}), \mathbf{p}) - f_i^L(s, X^B(s, \widehat{P}), \widehat{P}) \right| \mathrm{d}s.$$

To bound the difference in the initial conditions, we can add and subtract $x_{0,i}(\mathbf{p}_0^*)$, where $\mathbf{p}_0^* \in \arg\min_{\mathbf{p} \in \widehat{P}} x_{0,i}(\mathbf{p})$, to obtain the bound:

$$\begin{aligned} |x_{0,i}(\mathbf{p}) - x_{0,i}^L(\widehat{P})| &= |x_{0,i}(\mathbf{p}) - x_{0,i}(\mathbf{p}_0^*) + x_{0,i}(\mathbf{p}_0^*) - x_{0,i}^L(\widehat{P})|, \\ &\leq |x_{0,i}(\mathbf{p}) - x_{0,i}(\mathbf{p}_0^*)| + |x_{0,i}(\mathbf{p}_0^*) - x_{0,i}^L(\widehat{P})|, \end{aligned}$$

then use the local Lipschitz continuity of the initial condition (Assumption 3.3.2.2) and known convergence order $\beta_{\mathbf{x}_0} \geq 1$ of the inclusion function for the initial condition (Assumption 3.4.2). Since we have fixed a compact set P', there is a Lipschitz constant $L_{\mathbf{x}_0} \in \mathbb{R}_+$ and Hausdorff convergence prefactor $\widetilde{k}_0 \in \mathbb{R}_+$ such that

$$|x_{0,i}(\mathbf{p}) - x_{0,i}(\mathbf{p}_0^*)| + |x_{0,i}(\mathbf{p}_0^*) - x_{0,i}^L(\widehat{P})| \leq L_{\mathbf{x}_0} w\left(\widehat{P}\right) + \widetilde{k}_0 w\left(\widehat{P}\right)^{\beta_{\mathbf{x}_0}}.$$

Next, we bound the contribution from the vector field. Observe that

$$\begin{aligned} &\int_{t_0}^{t} \left| f_i(s, \mathbf{x}(s, \mathbf{p}), \mathbf{p}) - f_i^L(s, X^B(s, \widehat{P}), \widehat{P}) \right| \mathrm{d}s \\ &\quad = \int_{t_0}^{t} \left| f_i(s, \mathbf{x}(s, \mathbf{p}), \mathbf{p}) - f_i(s, \mathbf{z}^*(s), \mathbf{p}^*(s)) + f_i(s, \mathbf{z}^*(s), \mathbf{p}^*(s)) - f_i^L(s, X^B(s, \widehat{P}), \widehat{P}) \right| \mathrm{d}s, \\ &\quad \leq \int_{t_0}^{t} \left| f_i(s, \mathbf{x}(s, \mathbf{p}), \mathbf{p}) - f_i(s, \mathbf{z}^*(s), \mathbf{p}^*(s)) \right| + \left| f_i(s, \mathbf{z}^*(s), \mathbf{p}^*(s)) - f_i^L(s, X^B(s, \widehat{P}), \widehat{P}) \right| \mathrm{d}s, \end{aligned}$$

where, for any $t \in I$, $(\mathbf{z}^*(t), \mathbf{p}^*(t))$ is a solution of $\min_{(\mathbf{z},\mathbf{p}) \in X^B(t,\widehat{P}) \times \widehat{P}} f_i(t, \mathbf{z}, \mathbf{p})$. Then use the local Lipschitz continuity of the vector field (Assumption 3.3.2.2) and known convergence order $\beta_{\mathbf{f}} \geq 1$ of the inclusion function for the vector field (Assumption 3.4.2) with parameters

$(L_{\mathbf{f}}, \widetilde{k}) \in \mathbb{R}^2_+$:

$$\begin{aligned}
|x_i(t,\mathbf{p}) - x_i^L(t,\widehat{P})| &\leq L_{\mathbf{x}_0} w\Big(\widehat{P}\Big) + \widetilde{k}_0 w\Big(\widehat{P}\Big)^{\beta_{\mathbf{x}_0}} + \int_{t_0}^{t} L_{\mathbf{f}} \max\Big\{w\Big(\widehat{P}\Big), w\Big(X^B(s,\widehat{P})\Big)\Big\}\,\mathrm{d}s \\
&\quad + \int_{t_0}^{t} \widetilde{k} \max\Big\{w\Big(\widehat{P}\Big), w\Big(X^B(s,\widehat{P})\Big)\Big\}^{\beta_{\mathbf{f}}}\,\mathrm{d}s, \\
&\leq L_{\mathbf{x}_0} w\Big(\widehat{P}\Big) + \widetilde{k}_0 w\Big(\widehat{P}\Big)^{\beta_{\mathbf{x}_0}} + \int_{t_0}^{t} L_{\mathbf{f}} \Big[w\Big(\widehat{P}\Big) + w\Big(X^B(s,\widehat{P})\Big)\Big]\,\mathrm{d}s \\
&\quad + \int_{t_0}^{t} \widetilde{k} \Big[w\Big(\widehat{P}\Big) + w\Big(X^B(s,\widehat{P})\Big)\Big]^{\beta_{\mathbf{f}}}\,\mathrm{d}s. \qquad (3.11)
\end{aligned}$$

Since $\beta_{\mathbf{x}_0} \geq 1$ and $\beta_{\mathbf{f}} \geq 1$, there exist $k_0, k \in \mathbb{R}_+$ such that

$$\begin{aligned}
&\widetilde{k}_0 w\Big(\widehat{P}\Big)^{\beta_{\mathbf{x}_0}} \leq k_0 w\Big(\widehat{P}\Big) \quad \text{and} \\
&\widetilde{k} \Big[w\Big(\widehat{P}\Big) + w\Big(X^B(s,\widehat{P})\Big)\Big]^{\beta_{\mathbf{f}}} \leq k \Big[w\Big(\widehat{P}\Big) + w\Big(X^B(s,\widehat{P})\Big)\Big].
\end{aligned}$$

The values $k_0 = \widetilde{k}_0 w(P')^{\beta_{\mathbf{x}_0}-1}$ and $k = \widetilde{k}\left[w(P') + w(X^B(s,P'))\right]^{\beta_{\mathbf{f}}-1}$ are sufficient for any $\widehat{P} \in \mathbb{I}P'$. We use these k, k_0 and compute the contributions from the time-invariant part of the integrands in (3.11) to obtain:

$$\begin{aligned}
|x_i(t,\mathbf{p}) - x_i^L(t,\widehat{P})| \leq L_{\mathbf{x}_0} w\Big(\widehat{P}\Big) + k_0 w\Big(\widehat{P}\Big) + (t-t_0)(L_{\mathbf{f}} + k) w\Big(\widehat{P}\Big) + \\
\int_{t_0}^{t} (L_{\mathbf{f}} + k) w\Big(X^B(s,\widehat{P})\Big)\mathrm{d}s.
\end{aligned}$$

Then:

$$|x_i(t,\mathbf{p}) - x_i^L(t,\widehat{P})| \leq c_1 w\Big(\widehat{P}\Big) + c_2 \int_{t_0}^{t} w\Big(X^B(s,\widehat{P})\Big)\mathrm{d}s,$$

where $c_1 \equiv [L_{\mathbf{x}_0} + k_0 + (t_f - t_0)(L_{\mathbf{f}} + k)]$ and $c_2 \equiv (L_{\mathbf{f}} + k)$. We can also obtain the same bound for $|x_i(t,\mathbf{p}) - x_i^U(t,\widehat{P})|$:

$$|x_i(t,\mathbf{p}) - x_i^U(t,\widehat{P})| \leq c_1 w\Big(\widehat{P}\Big) + c_2 \int_{t_0}^{t} w\Big(X^B(s,\widehat{P})\Big)\mathrm{d}s.$$

Then note that

$$\begin{aligned} |x_i^U(t,\widehat{P}) - x_i^L(t,\widehat{P})| &= |x_i^U(t,\widehat{P}) - x_i(t,\mathbf{p}) + x_i(t,\mathbf{p}) - x_i^L(t,\widehat{P})|, \\ &\leq |x_i^U(t,\widehat{P}) - x_i(t,\mathbf{p})| + |x_i(t,\mathbf{p}) - x_i^L(t,\widehat{P})|, \\ &\leq 2c_1 w\Big(\widehat{P}\Big) + 2c_2 \int_{t_0}^{t} w\Big(X^B(s,\widehat{P})\Big)\mathrm{d}s. \end{aligned}$$

Next,

$$\begin{aligned} w\Big(X^B(t,\widehat{P})\Big) &= \max_{i\in\{1,\ldots,n_x\}} |x_i^U(t,\widehat{P}) - x_i^L(t,\widehat{P})| \\ &\leq 2c_1 w\Big(\widehat{P}\Big) + 2c_2 \int_{t_0}^{t} w\Big(X^B(s,\widehat{P})\Big)\mathrm{d}s, \end{aligned}$$

since the bound above does not depend on the particular $i \in \{1,\ldots,n_x\}$. We can apply the Gronwall-Bellman inequality to see that

$$w\Big(X^B(t,\widehat{P})\Big) \leq 2c_1 w\Big(\widehat{P}\Big)\exp\left(2c_2\int_{t_0}^{t}\mathrm{d}s\right) = 2c_1 w\Big(\widehat{P}\Big)\exp(2c_2(t-t_0)), \quad \forall t \in I.$$

Since $w(I) = t_f - t_0$ is finite,

$$w\Big(X^B(t,\widehat{P})\Big) \leq c_3 w\Big(\widehat{P}\Big), \quad \forall t \in I,$$

where $c_3 \equiv 2c_1 e^{2c_2 w(I)}$. Since $X^B(t,\widehat{P}) \supset \mathbf{x}(t,\widehat{P})$ for all $t \in I$,

$$d_H(X^B(t,\widehat{P}), \Box\mathbf{x}(t,\widehat{P})) \leq w\Big(X^B(t,\widehat{P})\Big) \leq c_3 w\Big(\widehat{P}\Big), \quad \forall t \in I.$$

Since $\widehat{P}$ was arbitrary, we have Hausdorff convergence in P' of order 1 with prefactor $\tau(t) \leq c_3$, $\forall t \in I$. □

Theorem 3.4.10. *If solutions exist to both the Harrison's method and naïve state bounding systems, then the state bounds resulting from Harrison's method have Hausdorff convergence in any compact $P' \subset P$ of order at least 1, uniformly on I.*

Proof. This follows directly from Proposition 3.4.7 and Theorem 3.4.9. □

The following proposition gives a situation in which bounds computed by Harrison's method can improve over time. Hypothesis (3.12) in the theorem is guaranteed to hold with $\alpha < 0$ for a linear dynamic system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ where at least one diagonal element of $\mathbf{A}$ is bounded above by $\alpha < 0$. Another sufficient condition to achieve (3.12) is: f_k is differentiable and $\frac{\partial f_k}{\partial x_k}(t, \mathbf{z}, \mathbf{p}) \leq \alpha < 0$, $\forall (\mathbf{z}, \mathbf{p}) \in X^B(t, \widehat{P}) \times \widehat{P}$. Specific examples where (3.12) holds with $\alpha < 0$ include mass-action chemical kinetic systems in which at least one species can be consumed by reaction, the (nonlinear) Lorenz equations with positive values of the parameters (σ, β), the Duffing equation with $\delta > 0$ (indicating positive damping of the harmonic oscillator), and the Lotka-Volterra model whenever $x_2 > 1$ or $x_1 < 1$. Strictly decreasing functions restricted to compact subsets of their domains also obey such a bound with $\alpha < 0$. For example, $\exp(-x)$.

Proposition 3.4.11. *Let $X^B(\cdot, \widehat{P})$ be a solution to the Harrison's method bounding system (Definition 3.4.5) for Problem 3.3.1. Let $P' \subset P$ be compact. Suppose for some $(t, k, \widehat{P}) \in I \times \{1, \ldots, n_x\} \times \mathbb{I}P'$, $\exists \alpha_k \in \mathbb{R}$ such that*

$$\begin{aligned} &f_k(t, \mathbf{z}^{(1)}, \mathbf{p}) - f_k(t, \mathbf{z}^{(2)}, \mathbf{p}) \leq \alpha_k (z_k^{(1)} - z_k^{(2)}), \\ &\qquad \forall (\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{p}) \in X^B(t, \widehat{P}) \times X^B(t, \widehat{P}) \times \widehat{P} : z_j^{(1)} = z_j^{(2)}, \forall j \neq k \text{ and } z_k^{(1)} \geq z_k^{(2)}. \end{aligned} \tag{3.12}$$

Then there exists $\tau_k \in \mathbb{R}_+$ such that the solution satisfies

$$\frac{dw(X_k^B)}{dt}(t, \widehat{P}) \leq \alpha_k w\Big(X_k^B(t, \widehat{P})\Big) + (L + \tau_k) \max\left\{\max_{i \neq k} w\Big(X_i^B(t, \widehat{P})\Big), w\Big(\widehat{P}\Big)\right\},$$

provided $L \in \mathbb{R}_+$ satisfies

$$\begin{aligned} |f_k(t, \mathbf{z}^{(1)}, \mathbf{p}^{(1)}) - f_k(t, \mathbf{z}^{(2)}, \mathbf{p}^{(2)})| \leq L \max\left\{\max_{i \neq k} |z_i^{(1)} - z_i^{(2)}|, \|\mathbf{p}^{(1)} - \mathbf{p}^{(2)}\|_\infty\right\}, \\ \forall (\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{p}^{(1)}, \mathbf{p}^{(2)}) \in (X^B(t, \widehat{P}))^2 \times (P')^2. \end{aligned}$$

If $\alpha_k < 0$, this bound can be negative. Furthermore, for the bounds given in Definition 3.4.3, it is not possible to obtain $\frac{dw(X_k^B)}{dt}(t, \widehat{P}) < 0$.

Proof. Choose any compact $P' \subset P$ containing the hypothesized $\widehat{P} \in \mathbb{I}P$. For this proof only, make the following definitions:

1. Let $F_{k,L/U}(t,\widehat{P})$ be the intervals given by the natural interval extensions

$$F_k(t,\mathcal{B}_k^{L/U}(X^B(t,\widehat{P})),\widehat{P}),$$

   where the notation L/U means the two definitions hold for L and U independently.

2. $f_k^{*,L}(t,\widehat{P}) \equiv \inf_{\mathbf{z}\in\mathcal{B}_k^L(X^B(t,\widehat{P})),\ \mathbf{p}\in\widehat{P}} f_k(t,\mathbf{z},\mathbf{p})$, and

3. $f_k^{*,U}(t,\widehat{P}) \equiv \sup_{\mathbf{z}\in\mathcal{B}_k^U(X^B(t,\widehat{P})),\ \mathbf{p}\in\widehat{P}} f_k(t,\mathbf{z},\mathbf{p})$.

Observe that

$$\begin{aligned}\frac{dw(X_k^B)}{dt}(t,\widehat{P}) &= F_{k,U}^U(t,\widehat{P}) - F_{k,L}^L(t,\widehat{P}),\\ &\leq \left(f_k^{*,U}(t,\widehat{P}) + w\Big(F_{k,U}(t,\widehat{P})\Big)\right) - \left(f_k^{*,L}(t,\widehat{P}) - w\Big(F_{k,L}(t,\widehat{P})\Big)\right),\\ &= f_k^{*,U}(t,\widehat{P}) - f_k^{*,L}(t,\widehat{P}) + w\Big(F_{k,U}(t,\widehat{P})\Big) + w\Big(F_{k,L}(t,\widehat{P})\Big),\end{aligned}$$

where the first equality is by the definition of $X_k^B(t,\widehat{P})$, the inequality follows from the facts $f_k^{*,L/U}(t,\widehat{P}) \in F_{k,L/U}(t,\widehat{P})$, and the second equality is by rearranging terms. Next, by Assumption 3.3.2.2 and the Weierstrass theorem (e.g., [24]) we know that the infimum and supremum of Definitions 2 and 3 above are attained. Assume they are attained at $(\mathbf{z}^{*,\min},\mathbf{p}^{*,\min})$ and $(\mathbf{z}^{*,\max},\mathbf{p}^{*,\max})$, respectively. Due to the $\mathcal{B}_k^{L/U}$ in the defining optimization problems, we have $z_k^{*,\min} = x_k^L(t,\widehat{P})$ and $z_k^{*,\max} = x_k^U(t,\widehat{P})$. Then,

$$\begin{aligned}&f_k^{*,U}(t,\widehat{P}) - f_k^{*,L}(t,\widehat{P})\\ &\quad= f_k(t,(z_1^{*,\max},\ldots,x_k^U(t,\widehat{P}),\ldots,z_{n_x}^{*,\max}),\mathbf{p}^{*,\max})\\ &\qquad- f_k(t,(z_1^{*,\min},\ldots,x_k^L(t,\widehat{P}),\ldots,z_{n_x}^{*,\min}),\mathbf{p}^{*,\min}).\end{aligned}$$

Note that $(t,(z_1^{*,\max},\ldots,x_k^L(t,\widehat{P}),\ldots,z_{n_x}^{*,\max}),\mathbf{p}^{*,\max})$ is guaranteed to be in the domain of f_k because for a solution to the Harrison's method bounding system to exist, the point

$$(t,(z_1,\ldots,x_k^L(t,\widehat{P}),\ldots,z_{n_x}),\mathbf{p}^{*,\max})$$

must be in the domain of f_k for any values of z_i, $i \neq k$ satisfying $z_i \in X_i^B(t, \widehat{P})$ for every $i \neq k$. Subtract and add

$$f_k(t, (z_1^{*,\max}, \ldots, x_k^L(t, \widehat{P}), \ldots, z_{n_x}^{*,\max}), \mathbf{p}^{*,\max}),$$

then use the α_k bound from Hypothesis 1 and local Lipschitz property from Assumption 3.3.2.2 to obtain:

$$\begin{aligned}
&f_k^{*,U}(t, \widehat{P}) - f_k^{*,L}(t, \widehat{P}) \\
&\quad = \Big[f_k(t, (z_1^{*,\max}, \ldots, x_k^U(t, \widehat{P}), \ldots, z_{n_x}^{*,\max}), \mathbf{p}^{*,\max}) \\
&\qquad - f_k(t, (z_1^{*,\max}, \ldots, x_k^L(t, \widehat{P}), \ldots, z_{n_x}^{*,\max}), \mathbf{p}^{*,\max}) \Big] \\
&\quad + \Big[f_k(t, (z_1^{*,\max}, \ldots, x_k^L(t, \widehat{P}), \ldots, z_{n_x}^{*,\max}), \mathbf{p}^{*,\max}) \\
&\qquad - f_k(t, (z_1^{*,\min}, \ldots, x_k^L(t, \widehat{P}), \ldots, z_{n_x}^{*,\min}), \mathbf{p}^{*,\min}) \Big], \\
&\quad \leq \alpha_k w\Big(X_k^B(t, \widehat{P}) \Big) + L \max \left\{ \max_{i \neq k} w\Big(X_i^B(t, \widehat{P}) \Big), w\Big(\widehat{P} \Big) \right\}.
\end{aligned}$$

Finally, apply linear Hausdorff convergence of the natural interval extension:

$$\begin{aligned}
&w\Big(F_{k,U}(t, \widehat{P}) \Big) \leq \tau_{k,U} \max \left\{ \max_{i \neq k} w\Big(X_i^B(t, \widehat{P}) \Big), w\Big(\widehat{P} \Big) \right\} \text{ and} \\
&w\Big(F_{k,L}(t, \widehat{P}) \Big) \leq \tau_{k,L} \max \left\{ \max_{i \neq k} w\Big(X_i^B(t, \widehat{P}) \Big), w\Big(\widehat{P} \Big) \right\}
\end{aligned}$$

to obtain:

$$\begin{aligned}
\frac{dw(X_k^B)}{dt}(t, \widehat{P}) &\leq \alpha_k w\Big(X_k^B(t, \widehat{P}) \Big) + (L + \tau_{k,U} + \tau_{k,L}) \max \left\{ \max_{i \neq k} w\Big(X_i^B(t, \widehat{P}) \Big), w\Big(\widehat{P} \Big) \right\}, \\
&= \alpha_k w\Big(X_k^B(t, \widehat{P}) \Big) + (L + \tau_k) \max \left\{ \max_{i \neq k} w\Big(X_i^B(t, \widehat{P}) \Big), w\Big(\widehat{P} \Big) \right\},
\end{aligned}$$

where $\tau_k = \tau_{k,L} + \tau_{k,U}$. To see that this bound can be negative, choose $\alpha_k < 0$ and take $w(X_k^B(t, \widehat{P}))$ arbitrarily large.

When the bounds of Definition 3.4.3 are used, $\frac{dw(X_k^B)}{dt}(t, \widehat{P}) = w(F_k^B(t, X^B(t, \widehat{P}), \widehat{P}))$, which must be nonnegative because $F_k^B(t, X^B(t, \widehat{P}), \widehat{P})$ is an interval. □

Definition 3.4.12. The upper right *Dini derivative* of $\psi : I \to \mathbb{R}$ is defined

$$(D_t^+\psi)(t) = \limsup_{h \to 0+} \frac{\psi(t+h) - \psi(t)}{h}, \quad \forall t \in I.$$

The Dini derivative is a generalization of a true derivative, and is useful for analyzing functions that are not necessarily differentiable.

Corollary 3.4.13. *Under the hypotheses of Proposition 3.4.11, suppose $X_k^B(t_0, \widehat{P})$ is so large that $w(X_k^B(t_0, \widehat{P})) > w(X_j^B(t_0, \widehat{P}))$, $\forall j \neq k$. Then, since we are using the ∞-norm for the diameter, $D_t^+ w(X^B)(t_0, \widehat{P}) = \frac{dw(X_k^B)}{dt}(t_0, \widehat{P}) < 0$. That is, the overall diameter of the state bounds can be decreasing at t_0.*

The following example gives a specific case where state bounds from Harrison's method satisfy $\frac{dw(X_i^B)}{dt}(t', \widehat{P}) < 0$ for some i and some t'.

Example 3.4.14. Consider the very simple chemical reaction A $\rightleftharpoons$ B, with the ODE model

$$\dot{x}_{\mathrm{A}} = -k_{\mathrm{f}} x_{\mathrm{A}} + k_{\mathrm{r}} x_{\mathrm{B}},$$

$$\dot{x}_{\mathrm{B}} = k_{\mathrm{f}} x_{\mathrm{A}} - k_{\mathrm{r}} x_{\mathrm{B}},$$

with $X_{\mathrm{A},0} \equiv [0.8, 1.2]$, $X_{\mathrm{B},0} \equiv [0.1, 0.1]$, $K_{\mathrm{f}} \equiv [15, 20]$, $K_{\mathrm{r}} \equiv [1, 5]$, $P \equiv k_{\mathrm{f}} \times k_{\mathrm{r}}$. Applying the rules of interval arithmetic, the vector field for the bounding system (Definition 3.4.5) for species A is:

$$\dot{x}_{\mathrm{A}}^L = \min\{-k_{\mathrm{f}}^L x_{\mathrm{A}}^L, -k_{\mathrm{f}}^U x_{\mathrm{A}}^L\} + \min\{k_{\mathrm{r}}^L x_{\mathrm{B}}^L, k_{\mathrm{r}}^L x_{\mathrm{B}}^U, k_{\mathrm{r}}^U x_{\mathrm{B}}^L, k_{\mathrm{r}}^U x_{\mathrm{B}}^U\},$$

$$\dot{x}_{\mathrm{A}}^U = \max\{-k_{\mathrm{f}}^L x_{\mathrm{A}}^U, -k_{\mathrm{f}}^U x_{\mathrm{A}}^U\} + \max\{k_{\mathrm{r}}^L x_{\mathrm{B}}^L, k_{\mathrm{r}}^L x_{\mathrm{B}}^U, k_{\mathrm{r}}^U x_{\mathrm{B}}^L, k_{\mathrm{r}}^U x_{\mathrm{B}}^U\}.$$

At the initial time, with the provided initial conditions and parameter ranges, we obtain $\dot{x}_{\mathrm{A}}^L(t_0) = -15.9 > -17.5 = \dot{x}_{\mathrm{A}}^U$. This means that at the initial time, the bounds for species A are becoming tighter, since $\frac{dw(X_{\mathrm{A}}^B)}{dt}(t_0, \widehat{P}) = \dot{x}_{\mathrm{A}}^U - \dot{x}_{\mathrm{A}}^L = -1.6$.

If we use the naïve state bounds of Definition 3.4.3, the vector field for the bounding system for species A are:

$$\begin{aligned}
\dot{\tilde{x}}_{\mathrm{A}}^{L} &= \min\{-k_{\mathrm{f}}^{L}x_{\mathrm{A}}^{L}, -k_{\mathrm{f}}^{L}x_{\mathrm{A}}^{U}, -k_{\mathrm{f}}^{U}x_{\mathrm{A}}^{L}, -k_{\mathrm{f}}^{U}x_{\mathrm{A}}^{U}\} + \min\{k_{\mathrm{r}}^{L}x_{\mathrm{B}}^{L}, k_{\mathrm{r}}^{L}x_{\mathrm{B}}^{U}, k_{\mathrm{r}}^{U}x_{\mathrm{B}}^{L}, k_{\mathrm{r}}^{U}x_{\mathrm{B}}^{U}\},\\
\dot{\tilde{x}}_{\mathrm{A}}^{U} &= \max\{-k_{\mathrm{f}}^{L}x_{\mathrm{A}}^{L}, -k_{\mathrm{f}}^{L}x_{\mathrm{A}}^{U}, -k_{\mathrm{f}}^{U}x_{\mathrm{A}}^{L}, -k_{\mathrm{f}}^{U}x_{\mathrm{A}}^{U}\} + \max\{k_{\mathrm{r}}^{L}x_{\mathrm{B}}^{L}, k_{\mathrm{r}}^{L}x_{\mathrm{B}}^{U}, k_{\mathrm{r}}^{U}x_{\mathrm{B}}^{L}, k_{\mathrm{r}}^{U}x_{\mathrm{B}}^{U}\},
\end{aligned}$$

which gives $\dot{\tilde{x}}_{\mathrm{A}}^{L}(t_0) = -23.9 < -1 = \dot{\tilde{x}}_{\mathrm{A}}^{U}(t_0)$, so that the bounds are becoming farther apart.

Next, we will build on Proposition 3.4.11 to give an integrated bound showing the time dependence of $w(X^B)$.

Theorem 3.4.15. *Let $X^B(\cdot, \widehat{P})$ be state bounds for the solution of Problem 3.3.1. Let $P' \subset P$ be compact. Suppose $\exists \boldsymbol{\alpha} \in \mathbb{R}^{n_x}$, such that for all $(t, k, \widehat{P}) \in I \times \{1, \dots, n_x\} \times \mathbb{I}P'$,*

$$\begin{aligned}
&f_k(t, \mathbf{z}^{(1)}, \mathbf{p}) - f_k(t, \mathbf{z}^{(2)}, \mathbf{p}) \leq \alpha_k(z_k^{(1)} - z_k^{(2)}),\\
&\qquad \forall(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{p}) \in \{(X^B(t, \widehat{P}))^2 \times \widehat{P} : z_j^{(1)} = z_j^{(2)}, \forall j \neq k \ \textit{and}\ z_k^{(1)} \geq z_k^{(2)}\}.
\end{aligned}$$

Let the matrix $\mathbf{S} \in \mathbb{R}^{n_x \times n_x}$ have elements given by:

$$S_{ij} = \begin{cases} \alpha_i & \textit{if } i = j,\\ L + \tau_i & \textit{if } i \neq j, \end{cases}$$

where each $\tau_i \in \mathbb{R}_+$ is the maximum of the corresponding value from Proposition 3.4.11 and for each $t \in I$, $L \in \mathbb{R}_+$ satisfies

$$\begin{aligned}
&\|\mathbf{f}(t, \mathbf{z}^{(1)}, \mathbf{p}^{(1)}) - \mathbf{f}(t, \mathbf{z}^{(2)}, \mathbf{p}^{(2)})\|_\infty \leq L\|(\mathbf{z}^{(1)}, \mathbf{p}^{(1)}) - (\mathbf{z}^{(2)}, \mathbf{p}^{(2)})\|_\infty,\\
&\qquad\qquad\qquad \forall(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{p}^{(1)}, \mathbf{p}^{(2)}) \in (X^B(t, \widehat{P}))^2 \times (P')^2.
\end{aligned}$$

Then, for each $(t, \widehat{P}) \in I \times \mathbb{I}P'$, the state bounds from Harrison's method (Definition 3.4.5) satisfy

1.

$$\frac{dw_V(X^B)}{dt}(t, \widehat{P}) \leq \mathbf{S}w_V\Big(X^B(t, \widehat{P})\Big) + (\boldsymbol{\tau} + L\mathbf{1})w\Big(\widehat{P}\Big),$$

where $\mathbf{1}$ *is a vector whose components are all 1.*

2. *If* $\mu_\infty(\mathbf{S}) \neq 0$*, then*

$$\begin{aligned} w\Big(X^B(t,\widehat{P})\Big) \leq &\left(w\Big(X^B(t_0,\widehat{P})\Big) + \frac{(L+\|\boldsymbol{\tau}\|_\infty)w(\widehat{P})}{\mu_\infty(\mathbf{S})}\right)\exp\left(\mu_\infty(\mathbf{S})(t-t_0)\right) \\ &- \frac{(L+\|\boldsymbol{\tau}\|_\infty)w(\widehat{P})}{\mu_\infty(\mathbf{S})}. \end{aligned} \tag{3.13}$$

3. *If instead* $\mu_\infty(\mathbf{S}) = 0$*, then*

$$w\Big(X^B(t,\widehat{P})\Big) \leq w\Big(X^B(t_0,\widehat{P})\Big) + \left[(L+\|\boldsymbol{\tau}\|_\infty)w\Big(\widehat{P}\Big)\right](t-t_0). \tag{3.14}$$

4. *If* $\alpha_i < -(n_x-1)|L+\tau_i|$*,* $\forall i$ *then* $\mu_\infty(\mathbf{S}) < 0$*, the state bounds can grow closer together as time increases, and the upper bound for* $w(X^B(t,\widehat{P}))$ *tends toward*

$$-\frac{(L+\|\boldsymbol{\tau}\|_\infty)w(\widehat{P}))}{\mu_\infty(\mathbf{S})}$$

as $t \to +\infty$*.*

Proof. Applying Proposition 3.4.11, we have for every k,

$$\frac{dw(X_k^B)}{dt}(t,\widehat{P}) \leq \alpha_k w\Big(X_k^B(t,\widehat{P})\Big) + (L+\tau_k)\max\left\{w\Big(\widehat{P}\Big), \max_{i\neq k} w\Big(X_i^B(t,\widehat{P})\Big)\right\}.$$

To obtain a linear bound, we can change the max operations to sums since all arguments of max are nonnegative:

$$\frac{dw(X_k^B)}{dt}(t,\widehat{P}) \leq \alpha_k w\Big(X_k^B(t,\widehat{P})\Big) + (L+\tau_k)\left(w\Big(\widehat{P}\Big) + \sum_{i\neq k} w\Big(X_i^B(t,\widehat{P})\Big)\right).$$

With

$$\mathbf{S} = \begin{bmatrix} \alpha_1 & L+\tau_1 & \cdots & \cdots & L+\tau_1 \\ L+\tau_2 & \alpha_2 & L+\tau_2 & \cdots & L+\tau_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & L+\tau_{n_x-1} \\ L+\tau_{n_x} & \cdots & L+\tau_{n_x} & L+\tau_{n_x} & \alpha_{n_x} \end{bmatrix},$$

it is clear that

$$\frac{dw_V(X^B)}{dt}(t,\widehat{P}) \leq \mathbf{S}w_V\Big(X^B(t,\widehat{P})\Big) + (L\mathbf{1}+\boldsymbol{\tau})w\Big(\widehat{P}\Big). \tag{3.15}$$

Next we will follow a similar line of reasoning to [190, (1.8)]. We write below the Dini derivative of the potentially nondifferentiable $\|w_V(X^B(t,\widehat{P}))\|_\infty$. First we note that $w_V(X^B(\cdot,\widehat{P}))$ is continuously differentiable because $X^B(\cdot,\widehat{P})$ is continuously differentiable, which is true because the vector fields defining $X^B(\cdot,\widehat{P})$ are locally Lipschitz.

$$\begin{aligned} D_t^+ w\Big(X^B(t,\widehat{P})\Big) &= D_t^+ \Big\| w_V\Big(X^B(t,\widehat{P})\Big)\Big\|_\infty, \\ &= \limsup_{h\to 0+} \frac{\Big\|w_V(X^B(t+h,\widehat{P}))\Big\|_\infty - \Big\|w_V(X^B(t,\widehat{P}))\Big\|_\infty}{h}, \\ &= \lim_{h\to 0+} \frac{\Big\|w_V(X^B(t,\widehat{P})) + h\frac{dw_V(X^B)}{dt}(t,\widehat{P})\Big\|_\infty - \Big\|w_V(X^B(t,\widehat{P}))\Big\|_\infty}{h}, \\ &\leq \lim_{h\to 0+} \frac{\Big\|w_V(X^B(t,\widehat{P})) + h\Big(\mathbf{S}w_V(X^B(t,\widehat{P})) + (L\mathbf{1}+\boldsymbol{\tau})w(\widehat{P})\Big)\Big\|_\infty - \Big\|w_V(X^B(t,\widehat{P}))\Big\|_\infty}{h}, \\ &\leq \lim_{h\to 0+} \frac{\|\mathbf{I}+h\mathbf{S}\|_\infty - 1}{h} \Big\|w_V\Big(X^B(t,\widehat{P})\Big)\Big\|_\infty + \Big\|(L\mathbf{1}+\boldsymbol{\tau})w\Big(\widehat{P}\Big)\Big\|_\infty, \\ &= \mu_\infty(\mathbf{S}) \Big\|w_V\Big(X^B(t,\widehat{P})\Big)\Big\|_\infty + (L+\|\boldsymbol{\tau}\|_\infty)w\Big(\widehat{P}\Big), \\ &= \mu_\infty(\mathbf{S})w\Big(X^B(t,\widehat{P})\Big) + (L+\|\boldsymbol{\tau}\|_\infty)w\Big(\widehat{P}\Big), \end{aligned}$$

where the first inequality holds since

$$\mathbf{0} \leq w_V\Big(X^B(t,\widehat{P})\Big) + h\frac{dw_V(X^B)}{dt}(t,\widehat{P}), \quad \text{for } h>0 \text{ sufficiently small,} \tag{3.16}$$

which we will justify next. Given any $\varepsilon > 0$, $\exists\delta > 0$ such that for all $h \in [0, \delta)$,

$$\begin{aligned}\mathbf{0} &\leq w_V\Big(X^B(t+h, \widehat{P})\Big),\\ &\leq w_V\Big(X^B(t, \widehat{P})\Big) + h\frac{dw_V(X^B)}{dt}(t, \widehat{P}) + \varepsilon.\end{aligned}$$

We used the fact that $w_V(X^B)(\cdot, \widehat{P})$ is differentiable; it need not be *continuously* differentiable. Parenthetically, $w_V(X^B)(\cdot, \widehat{P})$ is in fact continuously differentiable since it is the solution of an ODE with a locally Lipschitz vector field. Since we can take $\varepsilon > 0$ arbitrarily small and the inequalities are weak, we have (3.16).

Looking at the bound for $D_t^+ w(X^B(t, \widehat{P}))$, it is clear that if $\mu_\infty(\mathbf{S}) < 0$, then for $w(X^B(t, \widehat{P}))$ sufficiently large, the bounds grow closer together with time. By Proposition 3.2.6, if $\alpha_i < -\sum_{k \neq i} L + \tau_i$, $\forall i$ or equivalently if $\alpha_i < -(n_x - 1)(L + \tau_i)$, $\forall i$, then $\mu_\infty(\mathbf{S}) < 0$.

For naïve state bounds, the same interval objects are used in the construction of the vector fields for both the lower and upper bounds, so the rate of change for the lower bound must be smaller than that for the upper bound, and the bounds can never become closer together over time.

By integrating the bound for $D_t^+ w(X^B(t, \widehat{P}))$ using [79, Theorem 11], we obtain

$$\begin{aligned}w\Big(X^B(t, \widehat{P})\Big) &\leq w\Big(X^B(t_0, \widehat{P})\Big) + \int_{t_0}^{t} \mu_\infty(\mathbf{S}) w\Big(X^B(s, \widehat{P})\Big) + (L + \|\boldsymbol{\tau}\|_\infty) w\Big(\widehat{P}\Big) \mathrm{d}s,\\ &= w\Big(X^B(t_0, \widehat{P})\Big) + (L + \|\boldsymbol{\tau}\|_\infty) w\Big(\widehat{P}\Big)(t - t_0) + \int_{t_0}^{t} \mu_\infty(\mathbf{S}) w\Big(X^B(s, \widehat{P})\Big) \mathrm{d}s.\end{aligned}$$

If $\mu_\infty(\mathbf{S}) = 0$, we obtain (3.14) directly. If $\mu_\infty(\mathbf{S}) \neq 0$, we apply Lemma 3.2.4 with $\mu \equiv \mu_\infty(\mathbf{S})$, $\lambda_0 \equiv w(X^B(t_0, \widehat{P}))$, $\lambda_1 \equiv (L + \|\boldsymbol{\tau}\|_\infty) w(\widehat{P})$, and $x \equiv w(X^B(\cdot, \widehat{P}))$ to obtain (3.13). If $\mu_\infty(\mathbf{S}) < 0$, it is clear that the upper bound for $w(X^B(t, \widehat{P}))$ tends toward

$$-\frac{(L + \|\boldsymbol{\tau}\|_\infty) w(\widehat{P}))}{\mu_\infty(\mathbf{S})}$$

as $t \to +\infty$. □

See Example 3.6.1 for an application of this convergence bound and the corresponding convergence bound for the state relaxations for a chemical kinetics problem.

3.5 Bounds on the convergence order of state relaxations

In this section, we extend the work of Bompadre and Mitsos [32] to develop convergence-order bounds for state relaxations. The salient results are ultimately given in Theorems 3.5.9 and 3.5.17. To reach that end, we use the Gronwall-Bellman inequality, so we require several convergence bounds formulated in different terms than in previous literature. Specifically, we require $(1,2)$-convergence of the relaxation functions for $\mathbf{x}_0$ and $\mathbf{f}$ (Assumption 3.5.3.1). We show in §3.9 that Assumption 3.5.3 holds for relaxation functions generated using the natural McCormick extension [125, 170, 178].

Definition 3.5.1 (d_M)**.** Let $\mathcal{Y}, \mathcal{Z} \in \mathbb{MR}^n$. Define

$$d_M(\mathcal{Y}, \mathcal{Z}) = \max\left\{d_H(Y^B, Z^B), d_H(Y^C, Z^C)\right\}.$$

Lemma 3.5.2. *$\mathbb{MR}^n$ is a metric space when equipped with the metric d_M.*

Proof. See [170, §2.5.1]. □

Assumption 3.5.3. *The relaxation functions used for $\mathbf{f}$ and $\mathbf{x}_0$ of Problem 3.3.1:*

1. *have $(1,2)$-convergence on any interval subset of their domains and*

2. *are locally Lipschitz.*

If $\mathbf{f}$ and $\mathbf{x}_0$ are factorable in a certain sense (Definition 3.9.3) and satisfy Assumptions 3.9.12 and 3.9.13, then Assumption 3.5.3 holds for the natural McCormick extensions [125, 170, 178] of $\mathbf{f}$ and $\mathbf{x}_0$.

3.5.1 Methods for generating state relaxations

Next, we consider two different methods for generating state relaxations: relaxation-amplifying dynamics (RAD) [177] and relaxation-preserving dynamics (RPD) [174]. Both methods utilize an auxiliary ODE system that can be numerically integrated to generate state relaxations. We refer the reader to the paper on RPD [174], which compares the two methods from a theoretical standpoint. The full convergence analysis was deferred to the present thesis.

The Cut operator (defined below) is a key part of the theoretical development of generalized McCormick relaxations [170, 178], and it is used throughout the computational procedure in order to tighten the resulting relaxations and ensure that they are inclusion monotonic.

Definition 3.5.4 (Cut function [170])**.** Let

$$\mathrm{Cut} : \mathbb{MR}^n \to \mathbb{MR}^n : \mathcal{Z} \mapsto (Z^B, Z^B \cap Z^C).$$

Definition 3.5.5 (MC function)**.** Let

$$\begin{aligned}
&\widetilde{\square} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{IR}^n : (\mathbf{v}, \mathbf{w}) \mapsto \left[\mathbf{v} - \max\left\{\mathbf{0}, \frac{1}{2}(\mathbf{v} - \mathbf{w})\right\}, \mathbf{w} + \max\left\{\mathbf{0}, \frac{1}{2}(\mathbf{v} - \mathbf{w})\right\}\right],\\
&\widetilde{\cap} : \mathbb{IR}^n \times \mathbb{IR}^n \to \mathbb{IR}^n : ([\mathbf{x}^L, \mathbf{x}^U], [\widehat{\mathbf{x}}^L, \widehat{\mathbf{x}}^U]) \mapsto [\mathrm{mid}\{\mathbf{x}^L, \mathbf{x}^U, \widehat{\mathbf{x}}^L\}, \mathrm{mid}\{\mathbf{x}^L, \mathbf{x}^U, \widehat{\mathbf{x}}^U\}],\\
&\mathrm{MC} : \mathbb{R}^{4n} \to \mathbb{MR}^n : (\mathbf{x}^L, \mathbf{x}^U, \mathbf{x}^{cv}, \mathbf{x}^{cc}) \mapsto (\widetilde{\square}(\mathbf{x}^L, \mathbf{x}^U), \widetilde{\square}(\mathbf{x}^L, \mathbf{x}^U)\widetilde{\cap}\widetilde{\square}(\mathbf{x}^{cv}, \mathbf{x}^{cc})),
\end{aligned}$$

where mid returns the middle value of three scalars and operates on vectors componentwise.

The MC operator is similar to the Cut operator, but differs in the following way. Let $X^B, X^C \in \mathbb{IR}^n$ such that $X^B \cap X^C = \emptyset$. Let $(Y^B, Y^C) \equiv \mathrm{Cut}((X^B, X^C))$ and $(Z^B, Z^C) \equiv \mathrm{MC}(\mathbf{x}^L, \mathbf{x}^U, \mathbf{x}^{cv}, \mathbf{x}^{cc})$. Then $Y^C = \emptyset$ but $Z^C \neq \emptyset$. The MC operator is used in the construction of relaxations of the solutions of ODEs. It ensures that valid bounds and relaxations are constructed from input data. For example, the numerical integrator, during iterations, could attempt to converge with values of the state bounds or relaxations such that $z_i^L > z_i^U$ or $z_i^{cv} > z_i^{cc}$. The MC operator ensures that such all inputs from the ODE integrator yield valid relaxations for the vector field of the ODE for the relaxation system, and therefore valid relaxations of the solutions of the ODE.

The following restates a definition for convex and concave relaxations of the solution of an ODE from [177].

Definition 3.5.6 (Implementation of relaxation-amplifying dynamics (RAD) [177])**.** Let $\mathbf{f}$,

$\mathbf{x}_0$, and $\mathbf{x}$ be defined as in Problem 3.3.1. Let $\widehat{P} \in \mathbb{I}P$ and let $\widehat{\mathcal{P}}_{\mathbf{p}} \equiv (\widehat{P}, [\mathbf{p}, \mathbf{p}])$. RAD for $(\mathbf{x}^{cv}, \mathbf{x}^{cc})$ are given by the IVP in ODEs:

$$\begin{aligned}
\dot{\mathbf{x}}^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) &= \{\mathbf{f}\}^{cv}(t, \mathrm{MC}(\mathbf{x}^L(t, \widehat{P}), \mathbf{x}^U(t, \widehat{P}), \mathbf{x}^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}})), \widehat{\mathcal{P}}_{\mathbf{p}}),\\
\dot{\mathbf{x}}^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) &= \{\mathbf{f}\}^{cc}(t, \mathrm{MC}(\mathbf{x}^L(t, \widehat{P}), \mathbf{x}^U(t, \widehat{P}), \mathbf{x}^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}})), \widehat{\mathcal{P}}_{\mathbf{p}}),\\
\mathbf{x}^{cv}(t_0, \widehat{\mathcal{P}}_{\mathbf{p}}) &= \{\mathbf{x}_0\}^{cv}(\widehat{\mathcal{P}}_{\mathbf{p}}),\\
\mathbf{x}^{cc}(t_0, \widehat{\mathcal{P}}_{\mathbf{p}}) &= \{\mathbf{x}_0\}^{cc}(\widehat{\mathcal{P}}_{\mathbf{p}}),
\end{aligned}$$

for every $\widehat{P} \in \mathbb{I}P$ and every $(t, \mathbf{p}) \in I \times \widehat{P}$, where $\{g\}$ indicates the natural McCormick extension (Definition 3.9.11) of a function g and $X^B(\cdot, \widehat{P})$ are state bounds.

The RAD of Definition 3.5.6 provide state relaxations for Problem 3.3.1 [177, Theorem 4.1]. The natural McCormick extension can be evaluated computationally using the library `libMC` [130] or its successor `MC++` (`http://www3.imperial.ac.uk/people/b.chachuat/research`). Moreover, the constructed functions are locally Lipschitz on $\mathbb{M}I \times \mathbb{M}D \times \mathbb{M}P$.

Lemma 3.5.7. *For any $\widehat{P} \in \mathbb{I}P$, the solution of the RAD satisfies*

$$\begin{aligned}
&\mathbf{x}^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \geq \mathbf{x}^L(t, \widehat{P}), \quad \mathbf{x}^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \leq \mathbf{x}^U(t, \widehat{P}),\\
&\widetilde{\square}(\mathbf{x}^L(t, \widehat{P}), \mathbf{x}^U(t, \widehat{P})) = [\mathbf{x}^L(t, \widehat{P}), \mathbf{x}^U(t, \widehat{P})],\\
&\widetilde{\square}(\mathbf{x}^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}})) = [\mathbf{x}^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}})],
\end{aligned} \tag{3.17}$$

for every $(t, \mathbf{p}) \in I \times \widehat{P}$.

Proof. Fix any $\widehat{P} \in \mathbb{I}P$ and any $(t, \mathbf{p}) \in I \times \widehat{P}$. Since the state bounds and relaxations are both valid, we have

$$\mathbf{x}(t, \mathbf{p}) \in X^B(t, \widehat{P}) \cap X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}}).$$

If any of the claims in (3.17) fails to hold, then $X^B(t, \widehat{P}) \cap X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}}) = \emptyset$, which contradicts the assumption that a unique solution to Problem 3.3.1 exists (Assumption 3.3.2.1). □

Corollary 3.5.8. *For any $\widehat{P} \in \mathbb{I}P$, the solution of the RAD satisfies*

$$\begin{aligned}
&\mathrm{MC}(\mathbf{x}^L(t,\widehat{P}), \mathbf{x}^U(t,\widehat{P}), \mathbf{x}^{cv}(t,\widehat{\mathcal{P}}_\mathbf{p}), \mathbf{x}^{cc}(t,\widehat{\mathcal{P}}_\mathbf{p})),\\
&\quad = \mathrm{Cut}((X^B(t,\widehat{P}), X^C(t,\widehat{\mathcal{P}}_\mathbf{p}))), \quad \forall (t,\mathbf{p}) \in I \times \widehat{P}.
\end{aligned}$$

If the relaxation functions for both the initial condition and the vector field converge pointwise in P with order at least γ, Theorem 3.5.9, below, shows that RAD converge pointwise in P with order at least γ as well. Theorem 3.5.13 shows that RPD (Definition 3.5.11) also converge in P with order at least γ.

Theorem 3.5.9. *Consider the dynamic system of Problem 3.3.1. Let $P' \in \mathbb{I}P$. Assume that state bounds X^B with Hausdorff convergence in P' of order $\beta_\mathbf{x} \geq 1$, uniformly on I, are available. Under Assumption 3.5.3, the RAD (Definition 3.5.6), if a solution exists, have pointwise convergence in P' of order 2, uniformly on I.*

Proof. Choose any $\widehat{P} \in \mathbb{I}P$ and any $(t,\mathbf{p}) \in I \times \widehat{P}$. By Definition 3.5.6 and Corollary 3.5.8, the difference between the convex underestimator and the concave overestimator is:

$$\begin{aligned}
&x_i^{cc}(t,\widehat{\mathcal{P}}_\mathbf{p}) - x_i^{cv}(t,\widehat{\mathcal{P}}_\mathbf{p}) = x_{0,i}^{cc}(\widehat{\mathcal{P}}_\mathbf{p}) - x_{0,i}^{cv}(\widehat{\mathcal{P}}_\mathbf{p})\\
&+ \int_{t_0}^{t} \{f_i\}^{cc}(s, \mathrm{Cut}((X^B(s,\widehat{P}), X^C(s,\widehat{\mathcal{P}}_\mathbf{p}))), \widehat{\mathcal{P}}_\mathbf{p})\\
&\qquad - \{f_i\}^{cv}(s, \mathrm{Cut}((X^B(s,\widehat{P}), X^C(s,\widehat{\mathcal{P}}_\mathbf{p}))), \widehat{\mathcal{P}}_\mathbf{p})\, \mathrm{d}s,
\end{aligned}$$

for each i. Observe that without Cut, we have the inequality:

$$\begin{aligned}
&x_i^{cc}(t,\widehat{\mathcal{P}}_\mathbf{p}) - x_i^{cv}(t,\widehat{\mathcal{P}}_\mathbf{p}) \leq x_{0,i}^{cc}(\widehat{\mathcal{P}}_\mathbf{p}) - x_{0,i}^{cv}(\widehat{\mathcal{P}}_\mathbf{p})\\
&+ \int_{t_0}^{t} \{f_i\}^{cc}(s, (X^B(s,\widehat{P}), X^C(s,\widehat{\mathcal{P}}_\mathbf{p})), \widehat{\mathcal{P}}_\mathbf{p})\\
&\qquad - \{f_i\}^{cv}(s, (X^B(s,\widehat{P}), X^C(s,\widehat{\mathcal{P}}_\mathbf{p})), \widehat{\mathcal{P}}_\mathbf{p})\, \mathrm{d}s.
\end{aligned} \tag{3.18}$$

By Assumption 3.5.3, there exist $\tau_{0,1}, \tau_{0,2} > 0$ such that

$$\begin{aligned}
w\Big(X_{0,i}^C(\widehat{\mathcal{P}}_\mathbf{p})\Big) &\leq \tau_{0,1} w([\mathbf{p},\mathbf{p}]) + \tau_{0,2} w\Big(\widehat{P}\Big)^2,\\
&= \tau_{0,2} w\Big(\widehat{P}\Big)^2,
\end{aligned} \tag{3.19}$$

where the equality holds since $w([\mathbf{p},\mathbf{p}])=0$. By Assumption 3.5.3, $\exists \tau_1,\tau_2,\tau_3 \in \mathbb{R}_+$ such that the integrand in (3.18) is bounded above by

$$\begin{aligned} w\Big(F_i^C(s,\mathcal{X}(s,\widehat{\mathcal{P}}_{\mathbf{p}}),\widehat{\mathcal{P}}_{\mathbf{p}})\Big) &\leq \tau_1 \max\left\{w\Big(X^B(t,\widehat{P})\Big), w\Big(\widehat{P}\Big)\right\}^2 + \tau_2 \max\left\{w\Big(X^C(s,\widehat{\mathcal{P}}_{\mathbf{p}})\Big), w([\mathbf{p},\mathbf{p}])\right\}, \\ &\leq \tau_3 w\Big(\widehat{P}\Big)^2 + \tau_2 w\Big(X^C(s,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) \end{aligned}$$

where τ_1,τ_2,τ_3 do not depend on the particular values of s, $\widehat{P}$, or $\mathbf{p}$. The second line above holds since we have assumed state bounds X^B with linear Hausdorff convergence, uniformly on I and $w([\mathbf{p},\mathbf{p}])=0$. With these bounds on the initial condition and integrand, (3.18) gives

$$w\Big(X_i^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) \leq \tau_{0,2} w\Big(\widehat{P}\Big)^2 + \int_{t_0}^{t} \tau_3 w\Big(\widehat{P}\Big)^2 + \tau_2 w\Big(X^C(s,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\mathrm{d}s.$$

Repeating for each i and taking the max, we obtain:

$$w\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) \leq \tau_{0,2} w\Big(\widehat{P}\Big)^2 + \int_{t_0}^{t} \tau_3 w\Big(\widehat{P}\Big)^2 + \tau_2 w\Big(X^C(s,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\mathrm{d}s,$$

The first term in the integrand above is time-independent, so we have

$$\begin{aligned} w\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) &\leq \tau_{0,2} w\Big(\widehat{P}\Big)^2 + (t-t_0)\tau_3 w\Big(\widehat{P}\Big)^2 + \int_{t_0}^{t} \tau_2 w\Big(X^C(s,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\mathrm{d}s, \\ &\leq \tau_{0,2} w\Big(\widehat{P}\Big)^2 + \tau_4 w\Big(\widehat{P}\Big)^2 + \int_{t_0}^{t} \tau_2 w\Big(X^C(s,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\mathrm{d}s, \end{aligned}$$

where $\tau_4 \equiv (t_f - t_0)\tau_3$. Next, apply the Gronwall-Bellman inequality to obtain:

$$\begin{aligned} w\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) &\leq \Big(\tau_{0,2} w\Big(\widehat{P}\Big)^2 + \tau_4 w\Big(\widehat{P}\Big)^2\Big) \exp(\tau_2(t-t_0)), \\ &\leq \tau_5 w\Big(\widehat{P}\Big)^2 \exp(\tau_2(t_f-t_0)), \end{aligned}$$

where $\tau_5 = \tau_{0,2} + \tau_4$. Finally, since $\widehat{P} \in \mathbb{I}P$ and $\mathbf{p} \in \widehat{P}$ were arbitrary, we have

$$\sup_{\mathbf{p}\in\widehat{P}} w\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) \leq \tau_6 w\Big(\widehat{P}\Big)^2, \quad \forall (t,\widehat{P}) \in I \times \mathbb{I}P,$$

where $\tau_6 = \tau_5 \exp(\tau_2(t_f - t_0))$. □

Now, we move on to the improved nonlinear ODE relaxation theory, termed RPD. Relaxations calculated using the original nonlinear ODE relaxation theory, RAD, were observed to have poor empirical convergence and poor CPU times in global dynamic optimization. The idea of "flattening" in calculating the vector field for the bounding system was motivated by Harrison's method, but this very flattening destroyed convexity if $X^C \not\subset X^B$, necessitating a numerical integration scheme with event detection to ensure $X^C \subset X^B$ at all times.

The following definition differs from that of $\mathcal{B}_i^{L/U}$ in that the function returns a pair of vectors rather than an interval.

Definition 3.5.10. Let $\mathcal{R}_i^{cv} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n : (\mathbf{v}, \mathbf{w}) \mapsto (\mathbf{v}, \mathbf{w}')$, where $w'_k = w_k$ if $k \neq i$ and $w'_i = v_i$. Similarly, let $\mathcal{R}_i^{cc} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n : (\mathbf{v}, \mathbf{w}) \mapsto (\mathbf{v}', \mathbf{w})$, where $v'_k = v_k$ if $k \neq i$ and $v'_i = w_i$.

Definition 3.5.11 (Implementation of relaxation-preserving dynamics (RPD) [170, 174])**.** Let $\mathbf{f}$, $\mathbf{x}_0$, and $\mathbf{x}$ be defined as in Problem 3.3.1. Relaxation-preserving dynamics for $(\mathbf{x}^{cv}, \mathbf{x}^{cc})$ are given by the IVP in ODEs:

$$\dot{x}_i^{cv}(t, \widehat{\mathcal{P}}_\mathbf{p}) = \begin{cases} u_i(t, X^B(t, \widehat{P}), X^C(t, \widehat{\mathcal{P}}_\mathbf{p}), \widehat{\mathcal{P}}_\mathbf{p}) & \text{if } b_i^{cv} = 0, \\ \max\{\dot{x}_i^L(t, \widehat{P}), u_i(t, X^B(t, \widehat{P}), X^C(t, \widehat{\mathcal{P}}_\mathbf{p}), \widehat{\mathcal{P}}_\mathbf{p})\} & \text{if } b_i^{cv} = 1, \end{cases}$$

$$\dot{x}_i^{cc}(t, \widehat{\mathcal{P}}_\mathbf{p}) = \begin{cases} o_i(t, X^B(t, \widehat{P}), X^C(t, \widehat{\mathcal{P}}_\mathbf{p}), \widehat{\mathcal{P}}_\mathbf{p}) & \text{if } b_i^{cc} = 0, \\ \min\{\dot{x}_i^U(t, \widehat{P}), o_i(t, X^B(t, \widehat{P}), X^C(t, \widehat{\mathcal{P}}_\mathbf{p}), \widehat{\mathcal{P}}_\mathbf{p})\} & \text{if } b_i^{cc} = 1, \end{cases}$$

$$\mathbf{x}^{cv}(t_0, \widehat{\mathcal{P}}_\mathbf{p}) = \{\mathbf{x}_0\}^{cv}(\widehat{\mathcal{P}}_\mathbf{p}),$$

$$\mathbf{x}^{cc}(t_0, \widehat{\mathcal{P}}_\mathbf{p}) = \{\mathbf{x}_0\}^{cc}(\widehat{\mathcal{P}}_\mathbf{p}),$$

$$\forall (i, t, \mathbf{p}) \in \{1, \ldots, n_x\} \times (t_0, t_f] \times \widehat{P},$$

where

$$
\begin{aligned}
&u_i(t, X^B(t,\widehat{P}), X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}), \widehat{\mathcal{P}}_{\mathbf{p}}) \\
&\quad = \{f_i\}^{cv}(t, \mathrm{MC}(\mathbf{x}^L(t,\widehat{P}), \mathbf{x}^U(t,\widehat{P}), \mathcal{R}_i^{cv}(\mathbf{x}^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}}))), \widehat{\mathcal{P}}_{\mathbf{p}}), \\
&o_i(t, X^B(t,\widehat{P}), X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}), \widehat{\mathcal{P}}_{\mathbf{p}}) \\
&\quad = \{f_i\}^{cc}(t, \mathrm{MC}(\mathbf{x}^L(t,\widehat{P}), \mathbf{x}^U(t,\widehat{P}), \mathcal{R}_i^{cc}(\mathbf{x}^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}}))), \widehat{\mathcal{P}}_{\mathbf{p}}), \\
&\qquad\quad \forall (t, \mathbf{p}, i) \in (t_0, t_f] \times \widehat{P} \times \{1, \dots, n_x\},
\end{aligned}
$$

X^B are state bounds, b_i^{cv} and b_i^{cc} are Boolean variables satisfying

$$
b_i^{cv} = \begin{cases} 0 & \text{if } x_i^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) > x_i^L(t,\widehat{P}) \\ 1 & \text{if } x_i^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) \leq x_i^L(t,\widehat{P}) \end{cases}, \qquad b_i^{cc} = \begin{cases} 0 & \text{if } x_i^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) < x_i^U(t,\widehat{P}) \\ 1 & \text{if } x_i^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) \geq x_i^U(t,\widehat{P}) \end{cases},
$$

and $\{g\}$ indicates the natural McCormick extension (Definition 3.9.11) of a function g.

Theorem 3.5.12. *Let $\widehat{P} \in \mathbb{I}P$. If a solution of the RPD of Definition 3.5.11 exists, it provides valid state relaxations. Furthermore, those relaxations satisfy $\mathbf{x}^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) \in X^B(t,\widehat{P}),\ \forall (t,\mathbf{p}) \in I \times \widehat{P}$.*

Proof. The first claim is proven in [174, Theorem 3] and [170, Chapter 7]. The second claim is proven in [174, Lemma 1]. □

Theorem 3.5.13. *Under the same hypotheses as Theorem 3.5.9 and the additional assumption that a solution exists for RPD, RPD give pointwise convergence in any $P' \in \mathbb{I}P$ of order 2, uniformly on I.*

Proof. We will show that the hypotheses of Theorem 3.4.6 hold with the definitions

$$\begin{aligned}
\widetilde{\mathbf{v}}(t,\widehat{P}) &= \mathbf{x}^{cv,RAD}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\\
\widetilde{\mathbf{w}}(t,\widehat{P}) &= \mathbf{x}^{cc,RAD}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\\
\mathbf{v}(t,\widehat{P}) &= \mathbf{x}^{cv,RPD}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\\
\mathbf{w}(t,\widehat{P}) &= \mathbf{x}^{cc,RPD}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\\
&\forall (t,\mathbf{p}) \in I \times \widehat{P}.
\end{aligned} \tag{3.20}$$

We have

$$\begin{aligned}
\widetilde{u}_i(t,Z,\widehat{P}) &= \{f_i\}^{cv}(t,\mathrm{MC}(\mathbf{x}^L(t,\widehat{P}),\mathbf{x}^U(t,\widehat{P}),\mathbf{z}^L,\mathbf{z}^U),\widehat{\mathcal{P}}_{\mathbf{p}}),\\
\widetilde{o}_i(t,Z,\widehat{P}) &= \{f_i\}^{cc}(t,\mathrm{MC}(\mathbf{x}^L(t,\widehat{P}),\mathbf{x}^U(t,\widehat{P}),\mathbf{z}^L,\mathbf{z}^U),\widehat{\mathcal{P}}_{\mathbf{p}}),\\
u_i(t,Z,\widehat{P}) &\geq \{f_i\}^{cv}(t,\mathrm{MC}(\mathbf{x}^L(t,\widehat{P}),\mathbf{x}^U(t,\widehat{P}),\mathcal{R}_i^{cv}(\mathbf{z}^L,\mathbf{z}^U)),\widehat{\mathcal{P}}_{\mathbf{p}}),\\
o_i(t,Z,\widehat{P}) &\leq \{f_i\}^{cc}(t,\mathrm{MC}(\mathbf{x}^L(t,\widehat{P}),\mathbf{x}^U(t,\widehat{P}),\mathcal{R}_i^{cc}(\mathbf{z}^L,\mathbf{z}^U)),\widehat{\mathcal{P}}_{\mathbf{p}}),\\
&\forall (t,\widehat{P},Z) \in I \times \mathbb{I}P \times \mathbb{I}D \text{ and every } \mathbf{p} \in \widehat{P},
\end{aligned}$$

where the same state bounds $[\mathbf{x}^L,\mathbf{x}^U]$, are used for both types of relaxations.

Relation (3.7) holds since the values at the initial conditions satisfy $\mathbf{v}(t_0,\widehat{P}) = \widetilde{\mathbf{v}}(t_0,\widehat{P})$ and $\mathbf{w}(t_0,\widehat{P}) = \widetilde{\mathbf{w}}(t_0,\widehat{P})$, $\forall \widehat{P} \in \mathbb{I}P$ and all $\mathbf{p} \in \widehat{P}$. Relation (3.8) holds since the natural McCormick extension is inclusion monotonic, $[\mathcal{R}_i^{cv}(\mathbf{z}^L,\mathbf{z}^U)] \subset Z$, $\forall Z \in \mathbb{I}D$ and $u_i(t,Z,\widehat{P}) \geq \widetilde{u}_i(t,[\mathcal{R}_i^{cv}(\mathbf{z}^L,\mathbf{z}^U)],\widehat{P})$, $\forall (i,Z,\widehat{P}) \in \{1,\ldots,n_x\} \times \mathbb{I}D \times \mathbb{I}P$ and analogously for each $\mathcal{R}_i^{cc}$, o_i, $\widetilde{o}_i$. Relation (3.9) holds since the natural McCormick extension is inclusion monotonic. Therefore, we have $[\mathbf{v}(t,\widehat{P})),\mathbf{w}(t,\widehat{P})] \subset [\widetilde{\mathbf{v}}(t,\widehat{P}),\widetilde{\mathbf{w}}(t,\widehat{P})]$, $\forall (t,\widehat{P}) \in I \times \mathbb{I}P$ with $\mathbf{v},\mathbf{w},\widetilde{\mathbf{v}},\widetilde{\mathbf{w}}$ defined in (3.20) for any $\mathbf{p} \in \widehat{P}$. Since the relaxations from RPD are at least as strong as those from RAD, the RPD relaxations inherit the convergence properties of the RAD relaxations. □

The following theorem gives a result for state relaxations analogous to the result that Proposition 3.4.11 gives for state bounds.

Proposition 3.5.14. *Let X^C be state relaxations for Problem 3.3.1 from RPD (Defini-*

tion 3.5.11). Let $P' \in \mathbb{I}P$. Suppose the state bounds X^B have linear Hausdorff convergence in P', uniformly on I. Suppose for some $(t, k, \widehat{P}) \in I \times \{1, \dots, n_x\} \times \mathbb{I}P'$,

1. *$\exists \alpha_k \in \mathbb{R}$ such that*

$$\begin{aligned} &f_k(t, \mathbf{z}^{(1)}, \mathbf{p}) - f_k(t, \mathbf{z}^{(2)}, \mathbf{p}) \leq \alpha_k(z_k^{(1)} - z_k^{(2)}), \\ &\quad \forall (\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{p}) \in \{(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}}))^2 \times \widehat{P} : z_j^{(1)} = z_j^{(2)}, \forall j \neq k \text{ and } z_k^{(1)} \geq z_k^{(2)}\}. \end{aligned}$$

2. *$X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \subset D, \forall \mathbf{p} \in \widehat{P}$.*

Then, the state relaxations satisfy

$$\begin{aligned} \frac{dw(X_k^C)}{dt}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \leq \alpha_k w\Big(X_k^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) &+ \tau_{1,k} w\Big(\widehat{P}\Big)^{\min\{2, \gamma_{\mathbf{f}}\}} \\ &+ \tau_{2,k} \max_{i \neq k} w\Big(X_i^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big), \forall \mathbf{p} \in \widehat{P}. \end{aligned}$$

If $\alpha_k < 0$, this bound can be negative. Furthermore, for the RAD (Definition 3.5.6), it is not possible to obtain $\frac{dw(X_k^C)}{dt}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) < 0$.

Proof. Choose any $\mathbf{p} \in \widehat{P}$. By Definition 3.5.11,

$$\begin{aligned} &\dot{x}_k^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \geq \{f_k\}^{cv}(t, \mathrm{MC}(\mathbf{x}^L(t, \widehat{P}), \mathbf{x}^U(t, \widehat{P}), \mathcal{R}_k^{cv}(\mathbf{x}^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}}))), \widehat{\mathcal{P}}_{\mathbf{p}}) \quad \text{and} \\ &\dot{x}_k^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \leq \{f_k\}^{cc}(t, \mathrm{MC}(\mathbf{x}^L(t, \widehat{P}), \mathbf{x}^U(t, \widehat{P}), \mathcal{R}_k^{cv}(\mathbf{x}^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}), \mathbf{x}^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}}))), \widehat{\mathcal{P}}_{\mathbf{p}}). \end{aligned}$$

By Theorem 3.5.12 and the fact that the state bounds are valid, the MC operator gives the same result as the Cut operator for this system:

$$\begin{aligned} &\dot{x}_k^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \geq \{f_k\}^{cv}(t, \mathrm{Cut}((X^B(t, \widehat{P}), \mathcal{B}_k^L(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})))), \widehat{\mathcal{P}}_{\mathbf{p}}) \quad \text{and} \\ &\dot{x}_k^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \leq \{f_k\}^{cc}(t, \mathrm{Cut}((X^B(t, \widehat{P}), \mathcal{B}_k^U(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})))), \widehat{\mathcal{P}}_{\mathbf{p}}). \end{aligned}$$

Using the fact above and the fact that $x_k^{cc}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \geq x_k^{cv}(t, \widehat{\mathcal{P}}_{\mathbf{p}})$ for all $(t, \mathbf{p}) \in I \times \widehat{P}$, we have

$$\begin{aligned}\frac{dw(X_k^C)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) = \frac{d\left(x_k^{cc}-x_k^{cv}\right)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\\ &\le \{f_k\}^{cc}(t,\mathrm{Cut}((X^B(t,\widehat{P}),\mathcal{B}_k^U(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})))),\widehat{\mathcal{P}}_{\mathbf{p}})\\ &\quad - \{f_k\}^{cv}(t,\mathrm{Cut}((X^B(t,\widehat{P}),\mathcal{B}_k^L(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})))),\widehat{\mathcal{P}}_{\mathbf{p}}).\end{aligned}$$

Invoking the hypothesis that $X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}) \subset D, \forall \mathbf{p} \in \widehat{P}$, we subtract and add the quantities

$$\begin{aligned}&f_k(t,(x_1(t,\mathbf{p}),\ldots,x_k^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\ldots,x_{n_x}(t,\mathbf{p})),\mathbf{p}) \text{ and}\\ &f_k(t,(x_1(t,\mathbf{p}),\ldots,x_k^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\ldots,x_{n_x}(t,\mathbf{p})),\mathbf{p})\end{aligned}$$

to obtain

$$\begin{aligned}\frac{dw(X_k^C)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) &= \big[\{f_k\}^{cc}(t,\mathrm{Cut}((X^B(t,\widehat{P}),\mathcal{B}_k^U(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})))),\widehat{\mathcal{P}}_{\mathbf{p}})\\ &\quad - f_k(t,(x_1(t,\mathbf{p}),\ldots,x_k^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\ldots,x_{n_x}(t,\mathbf{p})),\mathbf{p})\big]\\ &+ \big[f_k(t,(x_1(t,\mathbf{p}),\ldots,x_k^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\ldots,x_{n_x}(t,\mathbf{p})),\mathbf{p})\\ &\quad - f_k(t,(x_1(t,\mathbf{p}),\ldots,x_k^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\ldots,x_{n_x}(t,\mathbf{p})),\mathbf{p})\big]\\ &+ \big[f_k(t,(x_1(t,\mathbf{p}),\ldots,x_k^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}}),\ldots,x_{n_x}(t,\mathbf{p})),\mathbf{p})\\ &\quad - \{f_k\}^{cv}(t,\mathrm{Cut}((X^B(t,\widehat{P}),\mathcal{B}_k^L(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})))),\widehat{\mathcal{P}}_{\mathbf{p}}))\big].\end{aligned}$$

Using the convergence bounds from Assumption 3.5.3, the linear convergence of X^B, and the assumptions on α_k, we have

$$\begin{aligned}\frac{dw(X_k^C)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) &\le \widetilde{\tau}_1 w\Big(\widehat{P}\Big)^2 + \widetilde{\tau}_2 w\Big(\mathcal{B}_k^U(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))\Big)\\ &\quad + \alpha_k w\Big(X_k^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\\ &\quad + \widetilde{\tau}_3 w\Big(\widehat{P}\Big)^2 + \widetilde{\tau}_4 w\Big(\mathcal{B}_k^L(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))\Big).\end{aligned} \tag{3.21}$$

Note that

$$w\Big(\mathcal{B}_k^L(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))\Big) = w\Big(\mathcal{B}_k^U(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))\Big) = \max_{i\neq k} w\Big(X_i^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big),$$

so that (3.21) becomes

$$\begin{aligned}\frac{dw(X_k^C)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) \leq \tau_{1,k} w\Big(\widehat{P}\Big)^2 &+ \tau_{2,k} \max_{i\neq k} w\Big(X_i^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\\ &+ \alpha_k w\Big(X_k^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big),\end{aligned}$$

where $\tau_{1,k} = \widetilde{\tau}_1 + \widetilde{\tau}_3$ and $\tau_{2,k} = \widetilde{\tau}_2 + \widetilde{\tau}_4$ are sufficient.

The constant α_k, which can be negative, multiplies $w(X_k^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))$, which must be nonnegative, so that there can be a negative contribution to the change in size of $w(X_k^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))$. If a system has $\alpha_k < 0$, then if $x_k^{cv}(t,\widehat{\mathcal{P}}_{\mathbf{p}})$ and $x_k^{cc}(t,\widehat{\mathcal{P}}_{\mathbf{p}})$ are sufficiently far apart relative to the diameters of $\widehat{P}$ and $X_i^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})$, $\forall i \neq k$, the overall sum can be negative.

Remark 3.5.15. For RAD, the natural McCormick extension of f_k takes the same arguments for the convex and concave relaxations, so that $\frac{dw(X_k^C)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) \geq 0$ always.

□

The following example shows a simple system in which relaxations can be improving at a particular time.

Example 3.5.16. Consider again the very simple chemical reaction A $\leftrightharpoons$ B, with the ODE model

$$\dot{x}_{\mathrm{A}} = -k_{\mathrm{f}}x_{\mathrm{A}} + k_{\mathrm{r}}x_{\mathrm{B}},$$

$$\dot{x}_{\mathrm{B}} = k_{\mathrm{f}}x_{\mathrm{A}} - k_{\mathrm{r}}x_{\mathrm{B}}.$$

Let $\mathcal{X}_{\mathrm{A},0} \equiv ([0.8,1.2],[0.8,1.2])$, $\mathcal{X}_{\mathrm{B},0} \equiv ([0.1,0.1],[0.1,0.1])$, $\mathcal{K}_{\mathrm{f}} \equiv ([15,20],[17.5,17.5])$,

$\mathcal{K}_{\mathrm{r}} \equiv ([1,5],[3,3])$, and $\mathcal{P} \equiv \mathcal{K}_{\mathrm{f}} \times \mathcal{K}_{\mathrm{r}}$. Using the facts

$$\begin{aligned}
&0 \leq x_{\mathrm{A}}^{L}(t_0), \quad 0 \leq x_{\mathrm{B}}^{L}(t_0), \quad 0 \leq x_{\mathrm{A}}^{cv}(t_0), \quad 0 \leq x_{\mathrm{B}}^{cv}(t_0),\\
&0 \leq k_{\mathrm{f}}^{L}, \quad 0 \leq k_{\mathrm{r}}^{L}, \quad k_{\mathrm{f}}^{cv} = k_{\mathrm{f}}^{cc}, \quad k_{\mathrm{r}}^{cv} = k_{\mathrm{r}}^{cc},
\end{aligned}$$

and applying the rules of the natural McCormick extension, the vector fields for the RPD for species A at t_0 are:

$$\begin{aligned}
u_{\mathrm{A}} &= -\min\{k_{\mathrm{f}}^{cc} x_{\mathrm{A}}^{L} + k_{\mathrm{f}}^{U} x_{\mathrm{A}}^{cv} - k_{\mathrm{f}}^{U} x_{\mathrm{A}}^{L}, k_{\mathrm{f}}^{cc} x_{\mathrm{A}}^{U} + k_{\mathrm{f}}^{L} x_{\mathrm{A}}^{cv} - k_{\mathrm{f}}^{L} x_{\mathrm{A}}^{U}\}\\
&\qquad + \max\{k_{\mathrm{r}}^{cv} x_{\mathrm{B}}^{L} + k_{\mathrm{r}}^{L} x_{\mathrm{B}}^{cv} - k_{\mathrm{r}}^{L} x_{\mathrm{B}}^{L}, k_{\mathrm{r}}^{cv} x_{\mathrm{B}}^{U} + k_{\mathrm{r}}^{U} x_{\mathrm{B}}^{cv} - k_{\mathrm{r}}^{U} x_{\mathrm{B}}^{U}\},\\
o_{\mathrm{A}} &= -\max\{k_{\mathrm{f}}^{cv} x_{\mathrm{A}}^{L} + k_{\mathrm{f}}^{L} x_{\mathrm{A}}^{cc} - k_{\mathrm{f}}^{L} x_{\mathrm{A}}^{L}, k_{\mathrm{f}}^{cv} x_{\mathrm{A}}^{U} + k_{\mathrm{f}}^{U} x_{\mathrm{A}}^{cc} - k_{\mathrm{f}}^{U} x_{\mathrm{A}}^{U}\}\\
&\qquad + \min\{k_{\mathrm{r}}^{cc} x_{\mathrm{B}}^{L} + k_{\mathrm{r}}^{U} x_{\mathrm{B}}^{cc} - k_{\mathrm{r}}^{U} x_{\mathrm{B}}^{L}, k_{\mathrm{r}}^{cc} x_{\mathrm{B}}^{U} + k_{\mathrm{r}}^{L} x_{\mathrm{B}}^{cc} - k_{\mathrm{r}}^{L} x_{\mathrm{B}}^{U}\},\\
\dot{x}_{\mathrm{A}}^{cv} &= \max\{u_{\mathrm{A}}, \dot{x}_{\mathrm{A}}^{L}\},\\
\dot{x}_{\mathrm{A}}^{cc} &= \min\{o_{\mathrm{A}}, \dot{x}_{\mathrm{A}}^{U}\},
\end{aligned}$$

where we have omitted the arguments to preserve readability and $(\dot{x}_{\mathrm{A}}^{L}, \dot{x}_{\mathrm{A}}^{U})$ are computed using Harrison's method as in Example 3.4.14. At the initial time, with the provided initial conditions and parameter ranges, we obtain

$$\begin{aligned}
u_{\mathrm{A}} &= -\min\{17.5 \cdot 0.8 + 20 \cdot 0.8 - 20 \cdot 0.8, 17.5 \cdot 1.2 + 15 \cdot 0.8 - 15 \cdot 1.2\}\\
&\qquad + \max\{3 \cdot 0.1 + 1 \cdot 0.1 - 1 \cdot 0.1, 3 \cdot 0.1 + 5 \cdot 0.1 - 5 \cdot 0.1\},\\
o_{\mathrm{A}} &= -\max\{17.5 \cdot 0.8 + 15 \cdot 1.2 - 15 \cdot 0.8, 17.5 \cdot 1.2 + 20 \cdot 1.2 - 20 \cdot 1.2\}\\
&\qquad + \min\{3 \cdot 0.1 + 5 \cdot 0.1 - 5 \cdot 0.1, 3 \cdot 0.1 + 1 \cdot 0.1 - 1 \cdot 0.1\},\\
\dot{x}_{\mathrm{A}}^{cv} &= \max\{-13.7, -15.9\} = -13.7,\\
\dot{x}_{\mathrm{A}}^{cc} &= \min\{-20.7, -17.5\} = -20.7.
\end{aligned}$$

This means that at the initial time, the relaxations for species A are becoming tighter, since $\frac{dw(X_{\mathrm{A}}^{C})}{dt}(t_0, \widehat{\mathcal{P}}_{\mathbf{p}}) = f_{\mathrm{A,RPD}}^{cc} - f_{\mathrm{A,RPD}}^{cv} = -7$.

If we use RAD, the vector fields at t_0 are:

$$\begin{aligned}
\dot{x}_{\mathrm{A}}^{cv} = &-\min\{k_{\mathrm{f}}^{cc}x_{\mathrm{A}}^{L} + k_{\mathrm{f}}^{U}\underbrace{x_{\mathrm{A}}^{cc}}_{} - k_{\mathrm{f}}^{U}x_{\mathrm{A}}^{L}, k_{\mathrm{f}}^{cc}x_{\mathrm{A}}^{U} + k_{\mathrm{f}}^{L}\underbrace{x_{\mathrm{A}}^{cc}}_{} - k_{\mathrm{f}}^{L}x_{\mathrm{A}}^{U}\} \\
&+ \max\{k_{\mathrm{r}}^{cv}x_{\mathrm{B}}^{L} + k_{\mathrm{r}}^{L}x_{\mathrm{B}}^{cv} - k_{\mathrm{r}}^{L}x_{\mathrm{B}}^{L}, k_{\mathrm{r}}^{cv}x_{\mathrm{B}}^{U} + k_{\mathrm{r}}^{U}x_{\mathrm{B}}^{cv} - k_{\mathrm{r}}^{U}x_{\mathrm{B}}^{U}\}, \\
\dot{x}_{\mathrm{A}}^{cc} = &-\max\{k_{\mathrm{f}}^{cv}x_{\mathrm{A}}^{L} + k_{\mathrm{f}}^{L}\underbrace{x_{\mathrm{A}}^{cv}}_{} - k_{\mathrm{f}}^{L}x_{\mathrm{A}}^{L}, k_{\mathrm{f}}^{cv}x_{\mathrm{A}}^{U} + k_{\mathrm{f}}^{U}\underbrace{x_{\mathrm{A}}^{cv}}_{} - k_{\mathrm{f}}^{U}x_{\mathrm{A}}^{U}\} \\
&+ \min\{k_{\mathrm{r}}^{cc}x_{\mathrm{B}}^{L} + k_{\mathrm{r}}^{U}x_{\mathrm{B}}^{cc} - k_{\mathrm{r}}^{U}x_{\mathrm{B}}^{L}, k_{\mathrm{r}}^{cc}x_{\mathrm{B}}^{U} + k_{\mathrm{r}}^{L}x_{\mathrm{B}}^{cc} - k_{\mathrm{r}}^{L}x_{\mathrm{B}}^{U}\},
\end{aligned}$$

where we have again omitted the arguments and the changed terms are highlighted using braces below. At the initial time, with the provided initial conditions and parameter ranges, we obtain

$$\begin{aligned}
\dot{x}_{\mathrm{A}}^{cv} = &-\min\{17.5\cdot 0.8 + 20\cdot\underbrace{1.2}_{} - 20\cdot 0.8, 17.5\cdot 1.2 + 15\cdot\underbrace{1.2}_{} - 15\cdot 1.2\} \\
&+ \max\{3\cdot 0.1 + 1\cdot 0.1 - 1\cdot 0.1, 3\cdot 0.1 + 5\cdot 0.1 - 5\cdot 0.1\}, \\
\dot{x}_{\mathrm{A}}^{cc} = &-\max\{17.5\cdot 0.8 + 15\cdot\underbrace{0.8}_{} - 15\cdot 0.8, 17.5\cdot 1.2 + 20\cdot\underbrace{0.8}_{} - 20\cdot 1.2\} \\
&+ \min\{3\cdot 0.1 + 5\cdot 0.1 - 5\cdot 0.1, 3\cdot 0.1 + 1\cdot 0.1 - 1\cdot 0.1\}, \\
\dot{x}_{\mathrm{A}}^{cv} = &-20.7, \\
\dot{x}_{\mathrm{A}}^{cc} = &-13.7.
\end{aligned}$$

so we have $\frac{dw(X_{\mathrm{A,RAD}}^{C})}{dt}(t_0) = f_{\mathrm{A,RAD}}^{cc} - f_{\mathrm{A,RAD}}^{cv} = +7$, so the relaxations are becoming farther apart with time.

Theorem 3.5.17. *Let X^C be state relaxations for Problem 3.3.1 from RPD (Definition 3.5.11). Let $P' \in \mathbb{I}P$. Let state bounds X^B have Hausdorff convergence in P' of order 1, uniformly on I. Suppose $\exists\boldsymbol{\alpha} \in \mathbb{R}^{n_x}$ such that for all $(t, k, \widehat{P}) \in I \times \{1, \ldots, n_x\} \times \mathbb{I}P$,*

$$\begin{aligned}
&f_k(t, \mathbf{z}^{(1)}, \mathbf{p}) - f_k(t, \mathbf{z}^{(2)}, \mathbf{p}) \leq \alpha_k(z_k^{(1)} - z_k^{(2)}), \\
&\quad \forall(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{p}) \in \{(X^B(t, \widehat{P}))^2 \times \widehat{P} : z_j^{(1)} = z_j^{(2)}, \forall j \neq k \textit{ and } z_k^{(1)} \geq z_k^{(2)}\}
\end{aligned}$$

and $X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \subset D, \forall \mathbf{p} \in \widehat{P}$. *Let the matrix* $\mathbf{S} \in \mathbb{R}^{n_x \times n_x}$ *have elements:*

$$S_{ij} = \begin{cases} \alpha_i & \textit{if } i = j, \\ \tau_{2,i} & \textit{if } i \neq j, \end{cases}$$

where each $\tau_{2,i}$ *is given by the maximum value achieved by the corresponding quantity in Proposition 3.5.14 over* $I \times \mathbb{I}P'$. *Then, the state relaxations satisfy*

1.

$$\frac{dw_V(X^C)}{dt}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \leq \mathbf{S}w_V\Big(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \boldsymbol{\tau}_1 w\Big(\widehat{P}\Big)^2,$$
$$\forall (t, \widehat{P}) \in I \times \mathbb{I}P' \textit{ and each } \mathbf{p} \in \widehat{P}.$$

2. *If* $\mu_\infty(\mathbf{S}) \neq 0$, *then*

$$w\Big(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) \leq \left(w\Big(X^C(t_0, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \frac{\|\boldsymbol{\tau}_1\|_\infty w(\widehat{P})^2}{\mu_\infty(\mathbf{S})}\right) \exp(\mu_\infty(\mathbf{S})(t - t_0)) - \frac{\|\boldsymbol{\tau}_1\|_\infty w(\widehat{P})^2}{\mu_\infty(\mathbf{S})},$$
$$\forall (t, \widehat{P}) \in I \times \mathbb{I}P' \textit{ and each } \mathbf{p} \in \widehat{P}. \tag{3.22}$$

3. *If instead* $\mu_\infty(\mathbf{S}) = 0$ *then*

$$w\Big(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) \leq w\Big(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \left[\|\boldsymbol{\tau}_1\|_\infty w\Big(\widehat{P}\Big)^2\right](t - t_0),$$
$$\forall (t, \widehat{P}) \in I \times \mathbb{I}P' \textit{ and each } \mathbf{p} \in \widehat{P}. \tag{3.23}$$

4. *If* $\alpha_i < -(n_x - 1)|\tau_{2,i}| \ \forall i$, *then* $\mu_\infty(\mathbf{S}) < 0$, *the relaxations grow closer together as time increases, and the upper bound for* $w(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}}))$ *tends toward*

$$-\frac{\|\boldsymbol{\tau}_1\|_\infty w(\widehat{P})^2}{\mu_\infty(\mathbf{S})}$$

as $t \to +\infty$.

For the RAD (Definition 3.5.6), the relaxations can never become closer together as time increases.

Proof. For every $(t, k, \widehat{P})$ and every $\mathbf{p} \in \widehat{P}$ we apply Proposition 3.5.14 to obtain

$$\frac{dw(X_k^C)}{dt}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \leq \alpha_k w\Big(X_k^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \tau_{1,k} w\Big(\widehat{P}\Big)^2 + \tau_{2,k} \max_{i \neq k} w\Big(X_i^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big).$$

To obtain a linear bound, we can change from $\max_{i \neq k}$ to $\sum_{i \neq k}$ since all arguments of max are nonnegative:

$$\frac{dw(X_k^C)}{dt}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \leq \alpha_k w\Big(X_k^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \tau_{1,k} w\Big(\widehat{P}\Big)^2 + \tau_{2,k} \sum_{i \neq k} w\Big(X_i^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big).$$

With

$$\mathbf{S} \equiv \begin{bmatrix} \alpha_1 & \tau_{2,1} & \cdots & \cdots & \tau_{2,1} \\ \tau_{2,2} & \alpha_2 & \tau_{2,2} & \cdots & \tau_{2,2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \tau_{2,n_x-1} \\ \tau_{2,n_x} & \cdots & \tau_{2,n_x} & \tau_{2,n_x} & \alpha_{n_x} \end{bmatrix},$$

it is clear that

$$\frac{dw_V(X^C)}{dt}(t, \widehat{\mathcal{P}}_{\mathbf{p}}) \leq \mathbf{S} w_V\Big(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \boldsymbol{\tau}_1 w\Big(\widehat{P}\Big)^2. \tag{3.24}$$

Next we will follow a similar line of reasoning to the proof of Theorem 3.4.15. We write the

Dini derivative of the potentially nondifferentiable $\|w_V(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))\|_\infty$ as

$$\begin{aligned}
D_t^+ w\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) &= D_t^+ \left\|w_V\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\right\|_\infty, \\
&= \limsup_{h\to 0+} \frac{\left\|w_V(X^C(t+h,\widehat{\mathcal{P}}_{\mathbf{p}}))\right\|_\infty - \left\|w_V(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))\right\|_\infty}{h}, \\
&= \limsup_{h\to 0+} \frac{\left\|w_V(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})) + h\frac{dw_V(X^C)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}})\right\|_\infty - \left\|w_V(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))\right\|_\infty}{h}, \\
&\leq \lim_{h\to 0+} \frac{\left\|w_V(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})) + h\left(\mathbf{S}w_V(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})) + \boldsymbol{\tau}_1 w(\widehat{P})^2\right)\right\|_\infty - \left\|w_V(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))\right\|_\infty}{h}, \\
&\leq \lim_{h\to 0+} \frac{\|\mathbf{I}+h\mathbf{S}\|_\infty - 1}{h} \left\|w_V\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\right\|_\infty + \left\|\boldsymbol{\tau}_1 w\Big(\widehat{P}\Big)^2\right\|_\infty, \\
&= \mu_\infty(\mathbf{S}) \left\|w_V\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big)\right\|_\infty + \|\boldsymbol{\tau}_1\|_\infty w\Big(\widehat{P}\Big)^2, \\
&= \mu_\infty(\mathbf{S}) w\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \|\boldsymbol{\tau}_1\|_\infty w\Big(\widehat{P}\Big)^2,
\end{aligned}$$

where the first inequality holds since

$$\mathbf{0} \leq w_V\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) + h\frac{dw_V(X^C)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}}), \quad \text{for } h>0 \text{ sufficiently small,} \tag{3.25}$$

which we will justify next. Given any $\varepsilon > 0$, $\exists\delta > 0$ such that for all $h \in [0,\delta)$,

$$\begin{aligned}
\mathbf{0} &\leq w_V\Big(X^C(t+h,\widehat{\mathcal{P}}_{\mathbf{p}})\Big), \\
&\leq w_V\Big(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})\Big) + h\frac{dw_V(X^C)}{dt}(t,\widehat{\mathcal{P}}_{\mathbf{p}}) + \varepsilon.
\end{aligned}$$

Since we can take $\varepsilon > 0$ arbitrarily small and the inequalities are weak, we have (3.25).

By the bound for $D_t^+ w(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))$, if $\mu_\infty(\mathbf{S}) < 0$ and $w(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}}))$ is sufficiently large, then $D_t^+ w(X^C(t,\widehat{\mathcal{P}}_{\mathbf{p}})) < 0$ so that the relaxations are becoming closer together with time. By Proposition 3.2.6, if $\alpha_i < -\sum_{k\neq i}|\tau_{2,i}|$, $\forall i$ or equivalently if $\alpha_i < -(n_x-1)|\tau_{2,i}|$, $\forall i$, then $\mu_\infty(\mathbf{S}) < 0$.

By integrating the bound for $D_t^+ w(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}}))$ using [79, Theorem 11],

$$\begin{aligned} w\Big(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) &\leq w\Big(X^C(t_0, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \int_{t_0}^{t} \mu_\infty(\mathbf{S}) w\Big(X^C(s, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) \\ &\quad + \|\boldsymbol{\tau}_1\|_\infty w\Big(\widehat{P}\Big)^2 \mathrm{d}s, \\ &= w\Big(X^C(t_0, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) + \|\boldsymbol{\tau}_1\|_\infty w\Big(\widehat{P}\Big)^2 (t - t_0) \\ &\quad + \int_{t_0}^{t} \mu_\infty(\mathbf{S}) w\Big(X^C(s, \widehat{\mathcal{P}}_{\mathbf{p}})\Big) \mathrm{d}s. \end{aligned}$$

If $\mu_\infty(\mathbf{S}) = 0$, we obtain (3.23) directly. If $\mu_\infty(\mathbf{S}) \neq 0$, we apply Lemma 3.2.4 with $\mu \equiv \mu_\infty(\mathbf{S})$, $\lambda_0 \equiv w(X^C(t_0, \widehat{\mathcal{P}}_{\mathbf{p}}))$, $\lambda_1 \equiv \|\boldsymbol{\tau}_1\|_\infty w(\widehat{P})^2$, and $x \equiv w(X^C(\cdot, \widehat{\mathcal{P}}_{\mathbf{p}}))$ to obtain (3.22). If $\mu_\infty(\mathbf{S}) < 0$, it is clear that the upper bound for $w(X^C(t, \widehat{\mathcal{P}}_{\mathbf{p}}))$ tends toward

$$-\frac{\|\boldsymbol{\tau}_1\|_\infty w(\widehat{P})^2}{\mu_\infty(\mathbf{S})}$$

as $t \to +\infty$.

For RAD, the same arguments are used in the construction of the vector fields for both the convex and concave relaxations, so the rate of change of the convex relaxation must be less than that of the concave relaxation, and the relaxations can never become closer together over time. □

3.5.2 Critical parameter interval diameter

Similar bounds to those derived above for addition, multiplication, and univariate composition can also be derived for interval arithmetic. Within any host interval P, a linear convergence bound can be derived for the Hausdorff convergence in P of the natural interval extension to the original function (see Proposition 3.9.9 or [136, Lemma 6.1]):

$$d_H(F^B(\widehat{P}), \mathbf{f}(\widehat{P})) \leq \tau_{\mathrm{IA}} w\Big(\widehat{P}\Big), \quad \forall \widehat{P} \in \mathbb{I}P.$$

A potentially higher-order bound can be derived for McCormick relaxations:

$$d_H(H_{\mathbf{f}}(\widehat{P}), \mathbf{f}(\widehat{P})) \leq \tau_{\mathrm{Mc}} w\Big(\widehat{P}\Big)^{\gamma_{\mathrm{Mc}}}, \quad \forall \widehat{P} \in \mathbb{I}P, \quad \gamma_{\mathrm{Mc}} \geq 1.$$

The natural McCormick extension [170, 178] intersects the relaxations with the bounds for the computation of each factor, so $H_{\mathbf{f}}(\widehat{P}) \subset F^B(\widehat{P}),\ \forall \widehat{P} \in \mathbb{I}P$. As a result,

$$d_H(H_{\mathbf{f}}(\widehat{P}), \mathbf{f}(\widehat{P})) \leq d_H(F^B(\widehat{P}), \mathbf{f}(\widehat{P})) \leq \tau_{\mathrm{IA}} w\Big(\widehat{P}\Big), \quad \forall \widehat{P} \in \mathbb{I}P.$$

If the convergence bound for the McCormick relaxations grows faster than linearly in $w(\widehat{P})$ while that for the interval extensions grows linearly, then the bound for the interval extensions will at some point be stronger than that for the McCormick relaxations, as we formalize below.

Proposition 3.5.18. *When the bounds for relaxations and intervals cross, $\tau_{\mathrm{Mc}} w(\widehat{P})^{\gamma_{Mc}} = \tau_{\mathrm{IA}} w(\widehat{P})$. Apart from $w(\widehat{P}) = 0$, this equation is satisfied when $w(\widehat{P}) = (\tau_{\mathrm{IA}}/\tau_{\mathrm{Mc}})^{1/(\gamma_{\mathrm{Mc}}-1)}$. For the case $\gamma_{Mc} = 2$, we have a critical parameter interval diameter equal to τ_{IA}/τ_{Mc}.*

Proof. Trivial. □

We refer to the critical parameter interval diameter as w_{crit}.

3.6 Numerical example and discussion

The following example will serve to illustrate the connection between the convergence-order bounds and results we see in practice with global dynamic optimization.

Example 3.6.1. Consider the three-species reversible series isomerization four-parameter estimation problem with error added for the first-order reversible chain reaction A $\leftrightharpoons$ B $\leftrightharpoons$ C from [67, 187, 201]. The ODE model is:

$$\begin{aligned}
&\dot{x}_1 = -p_1 x_1 + p_2 x_2,\\
&\dot{x}_2 = p_1 x_1 - (p_2 + p_3) x_2 + p_4 x_3,\\
&\dot{x}_3 = p_3 x_2 - p_4 x_3,\\
&\mathbf{x}(t_0) = (1, 0, 0),\\
&\mathbf{p} \in P = [0, 10]^2 \times [10, 50]^2,\\
&t \in [t_0, t_f] = [0, 1],
\end{aligned}$$

and it can be shown with viability theory [7] that the solution must satisfy $\mathbf{x}(t,\mathbf{p}) \in [0,1]^3$ for all $(t,\mathbf{p}) \in I \times P$.

For this problem, we have observed that RAD [170, 177] give empirical convergence behavior that is quadratic on short time scales, but linear on longer time scales (Figure 3-2). This is consistent with the convergence prefactor for the state bounds growing more slowly than that for the state relaxations. This behavior leads to a w_{crit} (Lemma 3.5.18) that decreases as time increases. On the other hand, RPD give quadratic empirical convergence for all time scales in the problem, indicating that the convergence prefactor for the state relaxations is growing at a similar rate to the convergence prefactor for the state bounds. With RAD, the matrix $\mathbf{S}_{\text{RAD}}$ in the bound from Theorem 3.5.17 is

$$\mathbf{S}_{\text{RAD}} = \begin{bmatrix} 10 & 10 & 0 \\ 10 & 70 & 60 \\ 0 & 60 & 60 \end{bmatrix},$$

so that $\mu_\infty[\mathbf{S}_{\text{RAD}}] = \max\{20, 140, 120\} = 140$. Therefore, the convergence bound for the RAD relaxations grows roughly as e^{140t}. For the RPD, we have

$$\mathbf{S}_{\text{RPD}} = \begin{bmatrix} 0 & 10 & 0 \\ 10 & -10 & 60 \\ 0 & 60 & -10 \end{bmatrix},$$

so that $\mu_\infty[\mathbf{S}_{\text{RPD}}] = \max\{10, 60, 50\} = 60$ so the convergence bound for RPD grows roughly as e^{60t}. Clearly, both of these bounds grow extremely quickly, but there is still a vast difference of $e^{140t} - e^{60t} \approx e^{140t}$ between the two bounds.

Since McCormick relaxations typically converge quadratically whereas natural interval extensions typically converge linearly, there is usually a *critical parameter interval diameter* (w_{crit}). For parameter intervals with smaller diameters than this, the quadratic convergence bound is the stronger of the two; otherwise, the linear convergence bound is the stronger. This means that in numerical examples, for large parameter intervals, the empirical convergence behavior will be linear whereas for small parameter intervals it will be quadratic.

Also, until parameter intervals become smaller than w_{crit}, clustering [62, 214] will ensue as though the convergence order is linear.

Using RAD, although there is a valid second-order convergence bound, empirical second-order convergence in dynamic optimization can be lost. If the prefactor in the convergence bound for the RAD state relaxations grows faster with time than the prefactor for the state bounds from Harrison's method, w_{crit} shrinks exponentially with time. For sufficiently long times, w_{crit} can be so small that quadratic convergence is never observed in practice.

We investigated empirically the time-dependence of the convergence behavior using a parameter estimation problem with the dynamic system of Example 3.6.1 with the experimental data from [67, Example 2]. We constructed a nested sequence of boxes in the decision space, all of which contained the global minimum. For the sum-of-squared-errors objective function h, we plotted the conservatism of the lower bounds ($\min_{\mathbf{p}\in\widehat{P}} h(\mathbf{p}) - \min_{\mathbf{p}\in\widehat{P}} h^{cv}(\widehat{\mathcal{P}}_{\mathbf{p}})$) versus the size of the intervals ($w(\widehat{P})$). We calculated the convex underestimator to the objective function calculated using both RAD and RPD. On log-log axes, we refer to the steepest slope of a line bounding this convergence behavior from above as the *empirical convergence order*. For the original parameter estimation problem, which had a time horizon of $[0, 1]$ and 20 data points (every 0.05 time units), we observed empirical convergence of order 1 with RAD. However, when we shortened the time horizon by a factor of 10, defining the objective function based only on the first two data points at times 0.05 and 0.1, we observed quadratic empirical convergence even with RAD. See Figure 3-2. We attribute this phenomenon to the conservatism growing much faster with time for RAD than for the Harrison bounds, so that at times later than 0.1, the linear convergence bound from the Harrison bounds is stronger for all $w(\widehat{P})$ considered, whereas up to a time of about 0.1, the quadratic convergence bound of the relaxations is stronger for $w(\widehat{P}) \leq 4$.

3.7 Conclusion

Convergence behavior of bounds and relaxations is pivotal to the success of deterministic global optimization algorithms. We proved that relaxations generated by relaxation-amplifying dynamics (RAD) [177] and relaxation-preserving dynamics (RPD) [174] both

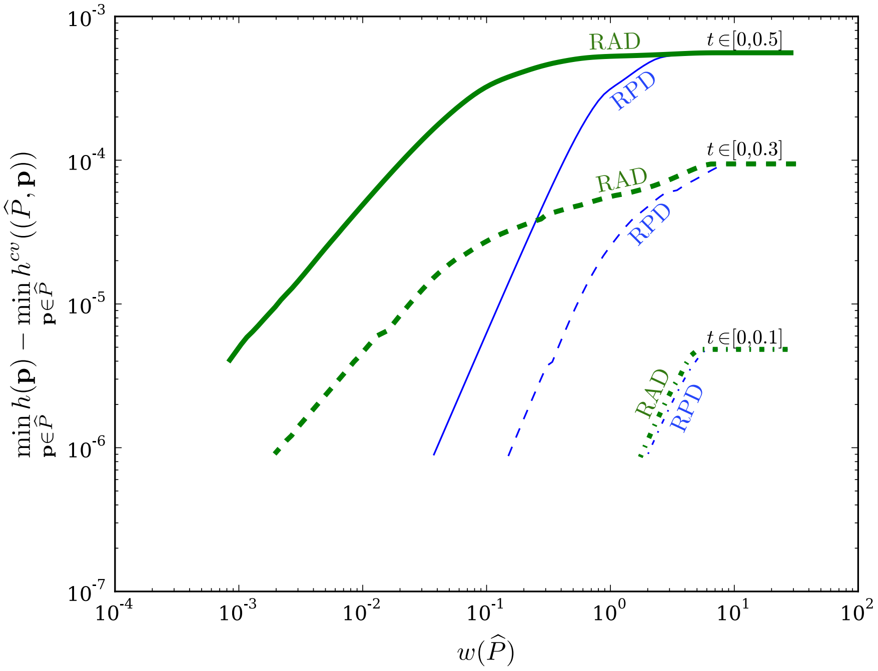


Figure 3-2: Empirically, the relaxations of the objective function for a test problem using RAD converge in P with order 2 at short integration times ($t \in [0, 0.1]$), but with order less than 1 at longer integration times. Relaxations based on RPD consistently converge in P with order 2.

obey second-order convergence bounds. We also illustrated how they can behave very differently from each other in practice, despite both obeying a second-order convergence bound. Numerically, we have found that relaxations generated using RAD give empirical first-order convergence in some test problems and empirical second-order convergence in others. On the other hand, relaxations generated using RPD give second-order empirical convergence most of the time. For RAD, the convergence-order bound for the state bounds grows more slowly with time than the convergence bound for the state relaxations, leading to a critical parameter interval diameter that decreases as time increases. This critical diameter is the locus of intersection of the first- and second-order convergence bounds. For intervals smaller than the critical diameter, the quadratic bound is dominant; otherwise the linear bound is. Because for RAD with Harrison bounds, the critical diameter decreases with time, longer time horizons increase the likelihood that RAD will display empirical first-order convergence even for very small intervals in parameter space. Even without the changeover in empirical convergence between first- and second-order at some critical parameter interval diameter, our analysis predicts that RPD could potentially perform much better, simply based on the potentially much smaller convergence prefactor. It was recently shown that a sufficiently small convergence prefactor can eliminate the cluster effect of global optimization [214].

One limitation of these results is that, as bounds that are always guaranteed to be valid, they represent the worst-case behavior. In much the same way, even if a theoretical convergence-order bound for method A is tighter than that for method B, in practice method A could still show inferior performance. This could occur, for example, if the convergence-order bound for method B were unnecessarily weak, or even if the convergence-order bound for method B held with equality for some problems, but just happened to be very weak for a particular problem instance. It is well-known that the simplex algorithm for linear programs shows worst-case exponential complexity, but it typically scales polynomially in practice. A smoothed analysis of the simplex algorithm was given by [191] to show why the simplex algorithm usually takes polynomial time. An analogous analysis may be possible for deterministic global dynamic optimization.

In the future, this analysis can be readily extended to incorporate additional bounding techniques. A relatively simple extension would show that if the bounds of [186] are used and *a priori* upper and lower bounds are known for every state variable, then the convergence-order bounds for the state bounds and relaxations should grow at most linearly in time. Due to the *a priori* known bounds on the solution of the ODE, the norm of the vector field has a time-invariant upper bound. This gives a convergence bound for the solution of the ODE that depends linearly on time rather than exponentially. The bounds of [172] exploit known invariant sets for the solution of the ODE, so a slightly more careful analysis may be required to develop a tight bound on their convergence behavior. It would also be interesting to analyze the convergence behavior of ODE bounding methods based on Taylor models [114, 116, 117, 163, 164], including the step in which the high-order Taylor model is bounded with a polynomial-time algorithm and the dependence of those convergence bounds on both the time horizon for the ODE and the diameter of the parameter interval. If other candidate bounding and relaxation methods for ODEs and DAEs are being developed, the present analysis framework could be used to estimate how they will behave in practice as well to examine whether and by what route the convergence order and prefactor can be improved.

3.8 Acknowledgments

We gratefully acknowledge funding from Novartis Pharmaceuticals. Joseph K. Scott contributed substantially to this work.

3.9 Supporting lemmas and proofs

3.9.1 Proof of Lemma 3.2.4

Proof. If $\mu = 0$, the result is trivial. Now consider $\mu \neq 0$. Define

$$v(s) \equiv \exp(\mu(t_0 - s)) \int_{t_0}^{s} \mu x(r) \mathrm{d}r, \quad \forall s \in I. \tag{3.26}$$

Differentiating gives

$$v'(s) = \mu \exp(\mu(t_0 - s)) \left(x(s) - \int_{t_0}^{s} \mu x(r) \mathrm{d}r \right), \quad \forall s \in I.$$

Since $\mu \neq 0$,

$$\frac{v'(s)}{\mu} = \underbrace{\exp(\mu(t_0 - s))}_{\geq 0} \underbrace{\left(x(s) - \int_{t_0}^{s} \mu x(r) \mathrm{d}r \right)}_{\leq \lambda_0 + \lambda_1(s - t_0)}, \quad \forall s \in I,$$

where the bound on the second term comes from (3.1). Therefore,

$$\frac{v'(s)}{\mu} \leq \exp(\mu(t_0 - s))(\lambda_0 + \lambda_1(s - t_0)), \quad \forall s \in I.$$

Note that $v(t_0) = 0$ and integrate:

$$\begin{aligned} \frac{v(t)}{\mu} &= \int_{t_0}^{t} \frac{v'(s)}{\mu} \mathrm{d}s \leq \int_{t_0}^{t} \exp(\mu(t_0 - s))(\lambda_0 + \lambda_1(s - t_0)) \mathrm{d}s, \\ &= \frac{\lambda_0 \mu + \lambda_1 - \exp(\mu(t_0 - t))(\lambda_0 \mu + \lambda_1 + \lambda_1 \mu(t - t_0))}{\mu^2}, \quad \forall t \in I. \end{aligned}$$

Substitute in the definition for v from (3.26):

$$\exp(\mu(t_0 - t)) \int_{t_0}^{t} x(s) \mathrm{d}s \leq \frac{\lambda_0 \mu + \lambda_1 - \exp(\mu(t_0 - t))(\lambda_0 \mu + \lambda_1 + \lambda_1 \mu(t - t_0))}{\mu^2}, \quad \forall t \in I.$$

Multiply by $\mu \exp(\mu(t - t_0))$:

$$\int_{t_0}^{t} \mu x(s) \mathrm{d}s \leq \frac{(\lambda_0 \mu + \lambda_1) \exp(\mu(t - t_0)) - (\lambda_0 \mu + \lambda_1 + \lambda_1 \mu(t - t_0))}{\mu}, \quad \forall t \in I.$$

Substitute the above inequality into (3.1) to obtain

$$\begin{aligned} x(t) &\leq \lambda_0 + \lambda_1(t - t_0) + \frac{(\lambda_0 \mu + \lambda_1) \exp(\mu(t - t_0)) - (\lambda_0 \mu + \lambda_1 + \lambda_1 \mu(t - t_0))}{\mu} \\ &= \left(\lambda_0 + \frac{\lambda_1}{\mu} \right) \exp(\mu(t - t_0)) - \frac{\lambda_1}{\mu}, \end{aligned}$$

for all $t \in I$. □

3.9.2 Proof of Theorem 3.4.6

Proof. The following argument follows a similar line of reasoning to [170, Corollary 3.3.6] and [170, Proof of Theorem 3.3.2], but is sufficiently different that we prove it in full.

Fix any $\widehat{P} \in \mathbb{I}P$. Since solutions to (3.5) and (3.6) are assumed to exist, we have

$$\begin{aligned}
&\mathbf{v}(t,\widehat{P}), \mathbf{w}(t,\widehat{P}), \widetilde{\mathbf{v}}(t,\widehat{P}), \widetilde{\mathbf{w}}(t,\widehat{P}) \in D, \quad \forall t \in I, \\
&\mathbf{v}(t,\widehat{P}) \leq \mathbf{w}(t,\widehat{P}) \quad \text{and} \quad \widetilde{\mathbf{v}}(t,\widehat{P}) \leq \widetilde{\mathbf{w}}(t,\widehat{P}), \quad \forall t \in I.
\end{aligned}$$

We need to prove that

$$\widetilde{\mathbf{v}}(t,\widehat{P}) \leq \mathbf{v}(t,\widehat{P}) \quad \text{and} \quad \mathbf{w}(t,\widehat{P}) \leq \widetilde{\mathbf{w}}(t,\widehat{P}), \quad \forall t \in I,$$

The initial conditions satisfy $[\widetilde{\mathbf{v}}(t_0,\widehat{P}), \widetilde{\mathbf{w}}(t_0,\widehat{P})] \supset [\mathbf{v}(t_0,\widehat{P}), \mathbf{w}(t_0,\widehat{P})]$. Suppose (to arrive at a contradiction) $\exists t \in I$ such that either $v_i(t,\widehat{P}) < \widetilde{v}_i(t,\widehat{P})$ or $w_i(t,\widehat{P}) > \widetilde{w}_i(t,\widehat{P})$ for at least one $i \in \{1,\ldots,n_x\}$ and define

$$t_1 \equiv \inf\{t \in I : v_i(t,\widehat{P}) < \widetilde{v}_i(t,\widehat{P}) \text{ or } w_i(t,\widehat{P}) > \widetilde{w}_i(t,\widehat{P}), \text{ for at least one } i\}.$$

Define

$$\boldsymbol{\delta} : I \to \mathbb{R}^{2n_x} : t \mapsto \boldsymbol{\delta}(t) \equiv (\widetilde{\mathbf{v}}(t,\widehat{P}) - \mathbf{v}(t,\widehat{P}), \mathbf{w}(t,\widehat{P}) - \widetilde{\mathbf{w}}(t,\widehat{P})).$$

By (3.7), $\boldsymbol{\delta}(t_0) \leq \mathbf{0}$ and $\exists t \in I$ such that $\delta_i(t) > 0$ for at least one i. Applying [170, Lemma 3.3.5], we obtain the following three facts.

1. $t_0 \leq t_1 < t_f$, $v_i(t,\widehat{P}) \geq \widetilde{v}_i(t,\widehat{P})$ and $w_i(t,\widehat{P}) \leq \widetilde{w}_i(t,\widehat{P})$, $\forall t \in [t_0,t_1]$.

2. At least one of the sets

$$\begin{aligned}
\mathcal{V}^L &\equiv \{i : \forall \gamma > 0, \exists t \in (t_1, t_1+\gamma] : v_i(t,\widehat{P}) < \widetilde{v}_i(t,\widehat{P})\}, \\
\mathcal{V}^U &\equiv \{i : \forall \gamma > 0, \exists t \in (t_1, t_1+\gamma] : w_i(t,\widehat{P}) > \widetilde{w}_i(t,\widehat{P})\},
\end{aligned}$$

is nonempty.

3. Let $t_4 \in (t_1, t_f]$, $\varepsilon > 0$, and $\beta \in L^1([t_1, t_4])$. Then there exists

   (a) $j \in \{1, \dots, n_x\}$,

   (b) a non-decreasing function $\rho \in \mathcal{AC}([t_1, t_4], \mathbb{R})$ satisfying

   $$0 < \rho(t) \leq \varepsilon, \quad \forall t \in [t_1, t_4] \quad \text{and} \quad \dot{\rho}(t) > |\beta(t)|\rho(t), \quad \text{a.e. } t \in [t_1, t_4], \tag{3.27}$$

   (c) numbers $t_2, t_3 \in [t_1, t_4]$ with $t_2 < t_3$ such that

   $$\widetilde{\mathbf{v}}(t, \widehat{P}) - \mathbf{1}\rho(t) < \mathbf{v}(t, \widehat{P}) \quad \text{and} \quad \mathbf{w}(t, \widehat{P}) < \widetilde{\mathbf{w}}(t, \widehat{P}) + \mathbf{1}\rho(t), \quad \forall t \in [t_2, t_3) \tag{3.28}$$

   and

   $$\begin{aligned} v_j(t_2, \widehat{P}) = \widetilde{v}_j(t_2, \widehat{P}), \quad v_j(t_3, \widehat{P}) = \widetilde{v}_j(t_3, \widehat{P}) - \rho(t_3), \\ \text{and} \quad v_j(t, \widehat{P}) < \widetilde{v}_j(t, \widehat{P}), \quad \forall t \in (t_2, t_3) \end{aligned} \tag{3.29}$$

   $$\Big(\text{or} \quad \begin{aligned} w_j(t_2, \widehat{P}) = \widetilde{w}_j(t_2, \widehat{P}), \quad w_j(t_3, \widehat{P}) = \widetilde{w}_j(t_3, \widehat{P}) + \rho(t_3), \\ \text{and} \quad w_j(t, \widehat{P}) > \widetilde{w}_j(t, \widehat{P}), \quad \forall t \in (t_2, t_3)\Big). \end{aligned} \tag{3.30}$$

Choose $\varepsilon > 0$ and $t_4 \in (t_1, t_f]$ sufficiently small that

$$[\widetilde{\mathbf{v}}(t, \widehat{P}), \widetilde{\mathbf{w}}(t, \widehat{P})] \subset [\widetilde{\mathbf{v}}(t_1, \widehat{P}), \widetilde{\mathbf{w}}(t_1, \widehat{P})] + [-\varepsilon\mathbf{1}, \varepsilon\mathbf{1}], \quad \forall t \in (t_1, t_4]$$

where

$$[\widetilde{\mathbf{v}}(t_1, \widehat{P}), \widetilde{\mathbf{w}}(t_1, \widehat{P})] + [-2\varepsilon\mathbf{1}, 2\varepsilon\mathbf{1}] \subset D.$$

and $\mathbf{1}$ is a vector with all components equal to 1. This is possible since (i) D is open, (ii) $[\widetilde{\mathbf{v}}(t, \widehat{P}), \widetilde{\mathbf{w}}(t, \widehat{P})] \subset D$, $\forall t \in I$ by existence of a solution to (3.6), and (iii) $\rho(t) < \varepsilon$, $\forall t \in [t_2, t_3)$. Let $L \in \mathbb{R}_+$ be the larger of the two Lipschitz constants for $\widetilde{\mathbf{u}}$ and $\widetilde{\mathbf{o}}$ on $[\widetilde{\mathbf{v}}(t_1, \widehat{P}), \widetilde{\mathbf{w}}(t_1, \widehat{P})] + [-2\varepsilon\mathbf{1}, 2\varepsilon\mathbf{1}]$. Let $\beta \equiv L$ and use the three facts from above.

Suppose that (3.29) holds (the proof is analogous if instead (3.30) holds). We know from (3.28) that

$$[\mathbf{v}(t,\widehat{P}),\mathbf{w}(t,\widehat{P})] \subset [\widetilde{\mathbf{v}}(t,\widehat{P}) - \rho(t)\mathbf{1}, \widetilde{\mathbf{w}}(t,\widehat{P}) + \rho(t)\mathbf{1}], \quad \forall t \in [t_2, t_3).$$

By (3.8), (3.9), and the inclusion above, we have

$$\begin{aligned} u_j(t,[\mathbf{v}(t,\widehat{P}),\mathbf{w}(t,\widehat{P})],\widehat{P}) &\geq \widetilde{u}_j(t,[\mathbf{v}(t,\widehat{P}),\mathbf{w}(t,\widehat{P})],\widehat{P}), \\ &\geq \widetilde{u}_j(t,[\widetilde{\mathbf{v}}(t,\widehat{P}) - \rho(t)\mathbf{1}, \widetilde{\mathbf{w}}(t,\widehat{P}) + \rho(t)\mathbf{1}],\widehat{P}), \quad \text{a.e. } t \in [t_2, t_3), \end{aligned} \tag{3.31}$$

where ε has already been chosen sufficiently small that $[\widetilde{\mathbf{v}}(t,\widehat{P}) - \rho(t)\mathbf{1}, \widetilde{\mathbf{w}}(t,\widehat{P}) + \rho(t)\mathbf{1}] \subset D$, $\forall t \in [t_2, t_3)$ (and therefore $[\widetilde{\mathbf{v}}(t,\widehat{P}) - \rho(t)\mathbf{1}, \widetilde{\mathbf{w}}(t,\widehat{P}) + \rho(t)\mathbf{1}] \in \mathbb{I}D$, $\forall t \in [t_2, t_3)$).

By the choice of L above, we have

$$|\widetilde{u}_j(t, Z^{(1)}, \widehat{P}) - \widetilde{u}_j(t, Z^{(2)}, \widehat{P})| \leq L d_H(Z^{(1)}, Z^{(2)}), \quad \forall Z^{(1)}, Z^{(2)} \in \mathbb{I}K. \tag{3.32}$$

with $K \equiv [\widetilde{\mathbf{v}}(t_1,\widehat{P}), \widetilde{\mathbf{w}}(t_1,\widehat{P})] + [-2\varepsilon\mathbf{1}, 2\varepsilon\mathbf{1}]$. Therefore,

$$\begin{aligned} \widetilde{u}_j(t,[\widetilde{\mathbf{v}}(t,\widehat{P}) - \rho(t)\mathbf{1}, \widetilde{\mathbf{w}}(t,\widehat{P}) + \rho(t)\mathbf{1}],\widehat{P}) \geq \widetilde{u}_j(t,[\widetilde{\mathbf{v}}(t,\widehat{P}),\widetilde{\mathbf{w}}(t,\widehat{P})],\widehat{P}) - L\rho(t), \\ \text{a.e. } t \in [t_2, t_3]. \end{aligned} \tag{3.33}$$

Combining (3.31) and (3.33),

$$u_j(t,[\mathbf{v}(t,\widehat{P}),\mathbf{w}(t,\widehat{P})],\widehat{P}) \geq \widetilde{u}_j(t,[\widetilde{\mathbf{v}}(t,\widehat{P}),\widetilde{\mathbf{w}}(t,\widehat{P})],\widehat{P}) - L\rho(t), \quad \text{a.e. } t \in [t_2, t_3].$$

Adding $\dot{\rho}(t)$ to both sides,

$$\begin{aligned} u_j(t,[\mathbf{v}(t,\widehat{P}),\mathbf{w}(t,\widehat{P})],\widehat{P}) + \dot{\rho}(t) &\geq \widetilde{u}_j(t,[\widetilde{\mathbf{v}}(t,\widehat{P}),\widetilde{\mathbf{w}}(t,\widehat{P})],\widehat{P}) - L\rho(t) + \dot{\rho}(t), \\ &\qquad \text{a.e. } t \in [t_2, t_3], \\ &\geq \widetilde{u}_j(t,[\widetilde{\mathbf{v}}(t,\widehat{P}),\widetilde{\mathbf{w}}(t,\widehat{P})],\widehat{P}), \quad \text{a.e. } t \in [t_2, t_3], \end{aligned}$$

where the second inequality follows from (3.27) with $\beta \equiv L \in \mathbb{R}_+$. By [170, Theorem 3.3.3], this implies that $(\widetilde{v}_j(\cdot, \widehat{P}) - v_j(\cdot, \widehat{P}) - \rho)$ is non-increasing on $[t_2, t_3]$, so that

$$\widetilde{v}_j(t_3, \widehat{P}) - v_j(t_3, \widehat{P}) - \rho(t_3) \leq \widetilde{v}_j(t_2, \widehat{P}) - v_j(t_2, \widehat{P}) - \rho(t_2),$$

but, by (3.29), this implies that $0 \leq -\rho(t_2)$, which contradicts (3.27). Since $\widehat{P} \in \mathbb{I}P$ was arbitrary, we have shown (3.10). □

3.9.3 $\mathcal{L}$-factorable functions

In the rest of the chapter, we use the notion of $\mathcal{L}$-factorable functions to analyze functions by factoring them into finite sequences of simple operations. Almost any function that can be represented finitely on a computer is $\mathcal{L}$-factorable provided the *library of univariate functions* $\mathcal{L}$ is sufficiently large. Roughly speaking, a function is $\mathcal{L}$-factorable if it can be broken down into a sequence of computational steps that use only univariate functions in $\mathcal{L}$ as well as the bivariate operations of addition and multiplication. When a function is $\mathcal{L}$-factorable, we can show that each cumulative mapping in the factored representation, including the overall function, has desirable properties as long as each univariate functions in the library $\mathcal{L}$ has certain properties. We use the concept of an $\mathcal{L}$-factorable function to show certain properties of the function itself, its natural interval extension, and its natural McCormick extension [170, 178]. These properties will satisfy the assumptions above (Assumptions 3.3.2.2, 3.4.2, and 3.5.3), so that the natural McCormick extension has the properties necessary for RAD [177] and RPD [174] to give quadratically-convergent relaxations of the solutions of ODEs. $\mathcal{L}$-factorable functions and other useful concepts in McCormick analysis were formalized in [170, Chapter 2], which unifies the ideas and notation of [178].

We will sometimes use the formal notation for a function as a triple (o, D, R), where D is the domain, R is the range, and o is a mapping from D into R. This will allow us to identify a function unambiguously while overloading standard functions to take interval or McCormick objects (the basic mathematical objects for the convex relaxation technique). For example, we can use exp for the real-valued function $\exp(p)$ or the interval-valued function $\exp(P)$. The elements of the set $\mathcal{L}$ are univariate functions $(u, B, \mathbb{R})$ satisfying $B \subset \mathbb{R}$. The elements of $\mathcal{L}$ represent functions such as $x \mapsto \sqrt{x}$, $x \mapsto x^n$, $x \mapsto \ln(x)$, or $x \mapsto \sin x$. Typically, $\mathcal{L}$ will also include the negative and reciprocal functions $x \mapsto -x$ and $x \mapsto 1/x$ so that subtraction and division can be achieved by combination with $(+, \mathbb{R}^2, \mathbb{R})$ and $(\times, \mathbb{R}^2, \mathbb{R})$.

Definition 3.9.1 ($\mathcal{L}$-computational sequence [170, Definition 2.2.1])**.** Let $n_i, n_o \in \mathbb{N}$. An $\mathcal{L}$-*computational sequence* with n_i inputs and n_o outputs is a pair $(\mathcal{S}, \pi_o)$:

1. $\mathcal{S}$ is a finite sequence $\{((o_k, B_k, \mathbb{R}), (\pi_k, \mathbb{R}^{k-1}, \mathbb{R}^{d_k}))\}_{k=n_i+1}^{n_f}$ with every element defined by one of the following options:

   (a) $(o_k, B_k, \mathbb{R})$ is either $(+, \mathbb{R}^2, \mathbb{R})$ or $(\times, \mathbb{R}^2, \mathbb{R})$ and $\pi_k : \mathbb{R}^{k-1} \to \mathbb{R}^2$ is defined by $\pi_k(\mathbf{v}) = (v_i, v_j)$ for some integers $i, j \in \{1, \ldots, k-1\}$.

   (b) $(o_k, B_k, \mathbb{R}) \in \mathcal{L}$ and $\pi_k : \mathbb{R}^{k-1} \to \mathbb{R}$ is defined by $\pi_k(\mathbf{v}) = v_i$ for some integer $i \in \{1, \ldots, k-1\}$.

2. $\pi_o : \mathbb{R}^{n_f} \to \mathbb{R}^{n_o}$ is defined by $\pi_o(\mathbf{v}) = (v_{i(1)}, \ldots, v_{i(n_o)})$ for some integers $i(1), \ldots, i(n_o) \in \{1, \ldots, n_f\}$.

A computational sequence defines a function $\mathbf{f}_{\mathcal{S}} : D_{\mathcal{S}} \subset \mathbb{R}^{n_i} \to \mathbb{R}^{n_o}$ in the following way.

Definition 3.9.2 ([170, Definition 2.2.2])**.** Let $(\mathcal{S}, \pi_o)$ be an $\mathcal{L}$-computational sequence with n_i inputs and n_o outputs. Define the *sequence of factors* $\{(v_k, D_k, \mathbb{R})\}_{k=1}^{n_f}$, with $D_k \subset \mathbb{R}^{n_i}$, where

1. For $k = 1, \ldots, n_i$, $D_k = \mathbb{R}^{n_i}$ and $v_k(\mathbf{p}) = p_k$, $\forall \mathbf{p} \in D_k$,

2. For $k = n_i + 1, \ldots, n_f$, $D_k = \{\mathbf{p} \in D_{k-1} : \pi_k(v_1(\mathbf{p}), \ldots, v_{k-1}(\mathbf{p})) \in B_k\}$, and $v_k(\mathbf{p}) = o_k(\pi_k(v_1(\mathbf{p}), \ldots, v_{k-1}(\mathbf{p})))$, $\forall \mathbf{p} \in D_k$.

The set $D_{\mathcal{S}} \equiv D_{n_f}$ is called the *natural domain* of $(\mathcal{S}, \pi_o)$ and the *natural function* $(\mathbf{f}_{\mathcal{S}}, D_{\mathcal{S}}, \mathbb{R}^{n_o})$ is defined by $\mathbf{f}_{\mathcal{S}}(\mathbf{p}) = \pi_o(v_1(\mathbf{p}), \ldots, v_{n_f}(\mathbf{p}))$, $\forall \mathbf{p} \in D_{\mathcal{S}}$.

Definition 3.9.3 ($\mathcal{L}$-factorable function [170, Definition 2.2.3])**.** Given any $D \subset \mathbb{R}^{n_i}$, a function $\mathbf{f} : D \to \mathbb{R}^{n_o}$ is said to be $\mathcal{L}$-*factorable* if there exists an $\mathcal{L}$-computational sequence

$(\mathcal{S}, \pi_o)$ with n_i inputs and n_o outputs such that the natural function $(\mathbf{f}_{\mathcal{S}}, D_{\mathcal{S}}, \mathbb{R}^{n_o})$ satisfies $D \subset D_{\mathcal{S}}$ and $\mathbf{f} = \mathbf{f}_{\mathcal{S}}|_D$.

With a sufficiently large library of univariate functions $\mathcal{L}$, an $\mathcal{L}$-factorable function can describe almost any function that can be represented finitely on a computer. See [170, Example 2.2.1].

The following assumption ensures that every $\mathcal{L}$-factorable function is locally Lipschitz [170, Theorem 2.5.26].

Assumption 3.9.4. *Every univariate function in $\mathcal{L}$ is locally Lipschitz.*

By [170, Theorem 2.5.26], Assumption 3.9.4 implies that any $\mathcal{L}$-factorable function is locally Lipschitz and therefore continuous on its domain. Furthermore, it implies that all of the factors in the $\mathcal{L}$-computational sequences for any $\mathcal{L}$-factorable function is locally Lipschitz.

3.9.4 Interval analysis

Definition 3.9.5 ([170, Definition 2.3.10])**.** For every $\mathcal{L}$-computational sequence $(\mathcal{S}, \pi_o)$ with n_i inputs and n_o outputs, define the *sequence of inclusion factors* $\{(V_k, \mathfrak{D}_k, \mathbb{IR})\}_{k=1}^{n_f}$ where

1. For all $k = 1, \ldots, n_i$, $\mathfrak{D}_k = \mathbb{IR}^{n_i}$ and $V_k(P) = P_k$, $\forall P \in \mathfrak{D}_k$,

2. For all $k = n_i + 1, \ldots, n_f$, $\mathfrak{D}_k = \{P \in \mathfrak{D}_{k-1} : \pi_k(V_1(P), \ldots, V_{k-1}(P)) \in \mathbb{I}B_k\}$ and $V_k(P) = o_k(\pi_k(V_1(P), \ldots, V_{k-1}(P)))$, $\forall P \in \mathfrak{D}_k$.

The *natural interval extension* of $(\mathcal{S}, \pi_o)$ is the function $(F_{\mathcal{S}}, \mathfrak{D}_{\mathcal{S}}, \mathbb{IR}^{n_o})$ defined by $\mathfrak{D}_{\mathcal{S}} \equiv \mathfrak{D}_{n_f}$ and $F_{\mathcal{S}}(P) = \pi_o(V_1(P), \ldots, V_{n_f}(P))$, $\forall P \in \mathfrak{D}_{\mathcal{S}}$.

The rules for the natural interval extension of o_k for addition, multiplication, and univariate composition were developed by [134] and are stated in this form in [170, Definition 2.3.6].

Assumption 3.9.6. *For every $(u, B, \mathbb{R}) \in \mathcal{L}$, the natural interval extension $(u, \mathbb{I}B, \mathbb{IR})$ is locally Lipschitz on $\mathbb{I}B$.*

By [170, Theorem 2.5.30], if Assumption 3.9.6 holds and $(\mathcal{S}, \pi_o)$ is an $\mathcal{L}$-computational sequence, then the natural interval extension $(F_{\mathcal{S}}, \mathfrak{D}_{\mathcal{S}}, \mathbb{IR}^{n_o})$ is locally Lipschitz on $\mathfrak{D}_{\mathcal{S}}$.

Proposition 3.9.7. *If $K \subset \mathbb{R}^n$ is compact, then $\mathbb{I}K$ is compact.*

Proof. Let $i_{\mathbb{R}} : \mathbb{IR}^n \to \{(\mathbf{a}, \mathbf{b}) \in \mathbb{R}^n \times \mathbb{R}^n : \mathbf{a} \leq \mathbf{b}\} : [\mathbf{z}^L, \mathbf{z}^U] \mapsto (\mathbf{z}^L, \mathbf{z}^U)$. It follows from the definition that $i_{\mathbb{R}}$ is bijective and isometric, i.e.,

$$d_H(Z_1, Z_2) = \|i_{\mathbb{R}}(Z_1) - i_{\mathbb{R}}(Z_2)\|_\infty.$$

Therefore, $i_{\mathbb{R}}^{-1}$ is a continuous function, so that it maps compact sets to compact sets. Then

$$\mathbb{I}K = i_{\mathbb{R}}^{-1}(\{(\mathbf{z}^L, \mathbf{z}^U) \in K^2 : \mathbf{z}^L \leq \mathbf{z}^U\})$$

is compact since $\{(\mathbf{z}^L, \mathbf{z}^U) \in K^2 : \mathbf{z}^L \leq \mathbf{z}^U\}$ is compact. □

Corollary 3.9.8. *Let $P \subset \mathbb{R}^n$ and $F : \mathbb{I}P \to \mathbb{IR}^m$ be locally Lipschitz on $\mathbb{I}P$. Then, for any compact $P' \subset P$, F is Lipschitz on $\mathbb{I}P'$.*

Proof. By Proposition 3.9.7, $\mathbb{I}P'$ is compact. The result follows from Proposition 3.2.2. □

In the following, given any function $(\mathbf{f}, D, \mathbb{R}^m)$, we use the notation $([\mathbf{f}], \mathfrak{D}, \mathbb{IR}^m)$ to denote its natural interval extension.

The following proposition shows how an interval extension being locally Lipschitz on $\mathbb{I}P$ guarantees Hausdorff convergence of order at least 1 on any compact $P' \subset P$.

Proposition 3.9.9. *Let $P \subset \mathbb{R}^{n_p}$ be nonempty. Let $\mathbf{f} : P \to \mathbb{R}^{n_x}$ be a continuous function, and let F be an inclusion function for $\mathbf{f}$ on $\mathbb{I}P$. If F is locally Lipschitz on $\mathbb{I}P$, then on any compact $P' \subset P$, it has Hausdorff convergence in P' of order 1 with prefactor $2L$, where $L \in \mathbb{R}_+$ is the Lipschitz constant for F on $\mathbb{I}P'$.*

Proof. By assumption, the inclusion function F is locally Lipschitz on $\mathbb{I}P$, which by Corollary 3.9.8 implies that for any compact $P' \subset P$ and any $j \in \{1, \ldots, n_x\}$, $\exists L \in \mathbb{R}_+$ such that

$$d_H(F_j(P^{(1)}), F_j(P^{(2)})) \leq L d_H(P^{(1)}, P^{(2)}), \quad \forall P^{(1)}, P^{(2)} \in \mathbb{I}P'.$$

Fix any $P^{(1)} \in \mathbb{I}P'$ and take $P^{(2)} = \{\mathbf{p}\}$ for some $\mathbf{p} \in P^{(1)}$. Then we know

$$d_H(F_j(P^{(1)}), F_j(\{\mathbf{p}\})) \leq L d_H(P^{(1)}, \{\mathbf{p}\}). \tag{3.34}$$

Next, observe that

$$d_H(P^{(1)}, \{\mathbf{p}\}) = \max_i \max\{|p_i^{(1),L} - p_i|, |p_i^{(1),U} - p_i|\} \leq w\Big(P^{(1)}\Big), \tag{3.35}$$

where, for each i, $p_i^{(1),L}$ and $p_i^{(1),U}$ are the bounds of p_i in $P^{(1)}$. Combining (3.34) and (3.35) we have

$$d_H(F_j(P^{(1)}), F_j(\{\mathbf{p}\})) \leq L w\Big(P^{(1)}\Big). \tag{3.36}$$

Since $F_j(\{\mathbf{p}\}) \subset F_j(P^{(1)})$ is a singleton,

$$0.5 w\Big(F_j(P^{(1)})\Big) \leq d_H(F_j(P^{(1)}), F_j(\{\mathbf{p}\})). \tag{3.37}$$

Combining (3.36) and (3.37), we have

$$w\Big(F_j(P^{(1)})\Big) \leq 2L w\Big(P^{(1)}\Big). \tag{3.38}$$

Since $f_j(P^{(1)}) \subset F_j(P^{(1)})$,

$$d_H(F_j(P^{(1)}), f_j(P^{(1)})) \leq w\Big(F_j(P^{(1)})\Big). \tag{3.39}$$

Combining (3.38) and (3.39),

$$d_H(F_j(P^{(1)}), f_j(P^{(1)})) \leq 2L w\Big(P^{(1)}\Big).$$

By Proposition 3.2.10, $d_H(F(P^{(1)}), \Box f(P^{(1)})) = \max_{j \in \{1,\ldots,n_x\}} d_H(F_j(P^{(1)}), \Box f_j(P^{(1)}))$, so

$$d_H(F(P^{(1)}), \Box \mathbf{f}(P^{(1)})) \leq 2L w\Big(P^{(1)}\Big).$$

Since $P^{(1)} \in \mathbb{I}P'$ was arbitrary, F has Hausdorff convergence in P' of order 1 with prefactor $2L$. $\square$

Corollary 3.9.10. *The natural interval extension has Hausdorff convergence of order 1 in any compact subset of $\mathfrak{D}_{\mathcal{S}}$.*

Proof. By Assumption 3.9.6 and [170, Theorem 2.5.30], the natural interval extension is locally Lipschitz on $\mathfrak{D}_{\mathcal{S}}$, and therefore it has Hausdorff convergence of order 1 in any compact subset of $\mathfrak{D}_{\mathcal{S}}$. $\square$

3.9.5 Natural McCormick extensions

The natural McCormick extension has its genesis in [125], but the statement below is from [170, Definition 2.4.31].

Definition 3.9.11 (Natural McCormick extension)**.** For every $\mathcal{L}$-computational sequence $(\mathcal{S}, \pi_o)$ with n_i inputs and n_o outputs, define the *sequence of relaxation factors* $\{(\mathcal{V}_k, \mathcal{D}_k, \mathbb{MR})\}_{k=1}^{n_f}$ where

1. for all $k = 1, \dots, n_i$, $\mathcal{D}_k = \mathbb{MR}^{n_i}$ and $\mathcal{V}_k(\mathcal{P}) = \mathcal{P}_k,\ \forall \mathcal{P} \in \mathcal{D}_k$,

2. for all $k = n_i + 1, \dots, n_f$, $\mathcal{D}_k = \{\mathcal{P} \in \mathcal{D}_{k-1} : \pi_k(\mathcal{V}_1(\mathcal{P}), \dots, \mathcal{V}_{k-1}(\mathcal{P})) \in \mathbb{M}B_k\}$ and $\mathcal{V}_k(\mathcal{P}) = o_k(\pi_k(\mathcal{V}_1(\mathcal{P})), \dots, \mathcal{V}_{k-1}(\mathcal{P})),\ \forall \mathcal{P} \in \mathcal{D}_k$.

The *natural McCormick extension* of $(\mathcal{S}, \pi_o)$ is the function $(\mathcal{F}_{\mathcal{S}}, \mathcal{D}_{\mathcal{S}}, \mathbb{MR}^{n_o})$ defined by $\mathcal{D}_{\mathcal{S}} \equiv \mathcal{D}_{n_f}$ and $\mathcal{F}(\mathcal{P}) = \pi_o(\mathcal{V}_1(\mathcal{P}), \dots, \mathcal{V}_{n_f}(\mathcal{P})),\ \forall \mathcal{P} \in \mathcal{D}_{\mathcal{S}}$. In the following, we will use the notation $(\{\mathbf{f}\}, \mathcal{D}, \mathbb{MR}^m)$ to denote a natural McCormick extension of $(\mathbf{f}, D, \mathbb{R}^m)$.

The rules for the natural McCormick extension of o_k for addition, multiplication, and univariate composition were developed by [125] and are stated in this form in [170, Definitions 2.4.18, 2.4.21, and 2.4.26]. The natural McCormick extension yields a relaxation function for $\mathbf{f}$ [32, 125, 170].

Assumption 3.9.12. *For every $(u, B, \mathbb{R}) \in \mathcal{L}$, the natural McCormick extension $(u, \mathbb{M}B, \mathbb{MR})$ is locally Lipschitz on $\mathbb{M}B$.*

By [170, Theorem 2.5.40], if Assumption 3.9.12 holds and $(\mathcal{S}, \pi_o)$ is an $\mathcal{L}$-computational sequence, then the natural McCormick extension $(\mathcal{F}_{\mathcal{S}}, \mathcal{D}_{\mathcal{S}}, \mathbb{MR}^{n_o})$ is locally Lipschitz on $\mathcal{D}_{\mathcal{S}}$.

Assumption 3.9.13. *For every $(u, B, \mathbb{R}) \in \mathcal{L}$, the natural McCormick extension $(u, \mathbb{M}B, \mathbb{MR})$ is known and converges pointwise in every $B' \in \mathbb{I}B$ with order 2.*

Assumption 3.9.13 is true of convex hulls. With Assumption 3.9.13 in place, we will see below that the natural McCormick extension of any $\mathcal{L}$-factorable function has $(1, 2)$-convergence. As a special case, it follows that the natural McCormick extension has pointwise convergence of order 2 in any $P' \in \mathbb{I}P$, as shown in [32].

3.9.6 Pointwise convergence bounds for $\mathcal{L}$-factorable functions

In this subsection, we develop a convergence bound for generalized McCormick relaxations in the form needed for the ODE relaxation theory [174, 177]. If (i) bounds and relaxations are available for some function $\mathbf{q} : \mathbb{R}^{n_p} \supset P \to X \subset \mathbb{R}^{n_x}$, (ii) $\phi : Y \to \mathbb{R}$ is $\mathcal{L}$-factorable, and (iii) $\mathbb{R}^{n_x} \supset Y \supset \mathbf{q}(P)$, then the generalized McCormick relaxation technique [178, Definition 15] allows us to obtain a relaxation function for some overall composite function $g \equiv \phi \circ \mathbf{q}$ in the following way. First, initialize factors $v_i^{L/U/cv/cc}$, $i = 1, \dots, n_x$ with the bounds and relaxations for each component of $\mathbf{q}$, then apply the rules of the natural McCormick extension to each factor in a factored representation of the outer function ϕ.

Proposition 3.9.14. *Let $F^C : \mathbb{M}P \to \mathbb{IR}^{n_x}$ be a relaxation function for the vector-valued function $\mathbf{f} : P \subset \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$ with pointwise convergence in P of order $\gamma_{\mathbf{f}}$. Then each component F_i^C of the relaxation function has pointwise convergence in P of order $\gamma_{f_i} \geq \gamma_{\mathbf{f}}$.*

Proof. For each i and each $\widehat{P} \in \mathbb{I}P$,

$$\sup_{\mathbf{p} \in \widehat{P}} w\Big(F_i^C(\widehat{\mathcal{P}}_{\mathbf{p}})\Big) \leq \sup_{\mathbf{p} \in \widehat{P}} w\Big(F^C(\widehat{\mathcal{P}}_{\mathbf{p}})\Big) \leq \tau_{\mathbf{f}} w\Big(\widehat{P}\Big)^{\gamma_{\mathbf{f}}}.$$

□

The following theorem extends [32, Theorem 1] to vector-valued functions.

Theorem 3.9.15. *Let $P \subset \mathbb{R}^{n_p}$. Suppose a relaxation function for a vector-valued function $\mathbf{f} : P \to \mathbb{R}^{n_x}$ has pointwise convergence in P of order γ. Then, the inclusion function $H_{\mathbf{f}}$ associated to the relaxation function has Hausdorff convergence in P of order $\beta \geq \gamma$.*

Proof. Fix any $\widehat{P} \in \mathbb{I}P$. By Proposition 3.2.10,

$$d_H(\square\mathbf{f}(\widehat{P}), H_{\mathbf{f}}(\widehat{P})) = \max_{i\in\{1,\ldots,n_x\}} d_H(f_i(\widehat{P}), H_{\mathbf{f},i}(\widehat{P})). \tag{3.40}$$

By Proposition 3.9.14, the relaxation function for each component f_i has pointwise convergence in P of order $\gamma_i \geq \gamma$. By [32, Theorem 1], the component of the inclusion function $H_{\mathbf{f},i}$ has Hausdorff convergence in P of order $\beta_{H_{\mathbf{f},i}} \geq \gamma_i$, so $\beta_{H_{\mathbf{f},i}} \geq \gamma$ and for each i there exists $\tau_{1,i} \in \mathbb{R}_+$ such that

$$d_H(f_i(\widehat{P}), H_{\mathbf{f},i}(\widehat{P})) \leq \tau_{1,i} w\left(\widehat{P}\right)^{\gamma},$$

so

$$\max_{i\in\{1,\ldots,n_x\}} d_H(f_i(\widehat{P}), H_{\mathbf{f},i}(\widehat{P})) \leq \max_{i\in\{1,\ldots,n_x\}} \tau_{1,i} w\left(\widehat{P}\right)^{\gamma}. \tag{3.41}$$

Taking (3.40) and (3.41) together we have

$$d_H(\square\mathbf{f}(\widehat{P}), H_{\mathbf{f}}(\widehat{P})) = \max_{i\in\{1,\ldots,n_x\}} d_H(f_i(\widehat{P}), H_{\mathbf{f},i}(\widehat{P}) \leq \tau w(P)^{\gamma},$$

for some $\tau \in \mathbb{R}_+$. □

3.9.7 (1,2)-Convergence of natural McCormick extensions

Along with linear convergence for the interval bounding method, $(1,2)$-convergence is easily composable; i.e.,

$$w(F^B(X^B)) \leq C_0 w(X^B), \tag{3.42}$$

$$w(\mathcal{F}(\mathcal{X})) \leq C_1 w(\mathcal{X}) + C_2 w(X^B)^2, \tag{3.43}$$

and

$$w(G^B(X^B)) \leq D_0 w(X^B), \tag{3.44}$$

$$w(\mathcal{G}(\mathcal{X})) \leq D_1 w(\mathcal{X}) + D_2 w(X^B)^2, \tag{3.45}$$

imply that

$$w(G^B \circ F^B(X^B)) \leq D_0 \tau_0 w(X^B), \tag{3.46}$$

$$w(\mathcal{G} \circ \mathcal{F}(\mathcal{X})) \leq D_1 w(\mathcal{F}(\mathcal{X})) + D_2 w(F^B(X^B))^2, \tag{3.47}$$

$$\leq D_1(C_1 w(\mathcal{X}) + C_2 w(X^B)^2) + D_2(\tau_0 w(X^B))^2, \tag{3.48}$$

$$\leq D_1 C_1 w(\mathcal{X}) + (D_1 C_2 + D_2 \tau_0^2) w(X^B)^2. \tag{3.49}$$

By this argument, it suffices to show that the basic McCormick operations are each $(1,2)$-convergent McCormick extensions of the corresponding real operations.

3.9.7.1 Addition

Definition 3.9.16. Define $(+, \mathbb{MR}^2, \mathbb{MR})$ by

$$+(\mathcal{X}, \mathcal{Y}) = \mathcal{X} + \mathcal{Y} = (X^B + Y^B, (X^B \cap X^C) + (Y^B \cap Y^C)). \tag{3.50}$$

Lemma 3.9.17. *$+(\mathcal{X}, \mathcal{Y})$ is (1,2)-convergent on $\mathbb{MR}^2$.*

Proof.

$$w(\mathcal{X} + \mathcal{Y}) = w(\mathrm{Enc}(\mathcal{X} + \mathcal{Y})), \tag{3.51}$$

$$= w((X^B \cap X^C) + (Y^B \cap Y^C)), \tag{3.52}$$

$$= w(X^B \cap X^C) + w(Y^B \cap Y^C), \tag{3.53}$$

$$= w(\mathcal{X}) + w(\mathcal{Y}), \tag{3.54}$$

$$\leq 2 \max\left(w(\mathcal{X}), w(\mathcal{Y})\right), \tag{3.55}$$

$$= 2w((\mathcal{X}, \mathcal{Y})). \tag{3.56}$$

Thus, $\tau_1 = 2$ and $\tau_2 = 0$. □

3.9.7.2 Multiplication

Definition 3.9.18. Define $(\times, \mathbb{MR}^2, \mathbb{MR})$ by

$$\times(\mathcal{X}, \mathcal{Y}) = \mathcal{X}\mathcal{Y} = (X^B Y^B, [z^{cv}, z^{cc}]), \tag{3.57}$$

where

$$z^{cv} = \max\left(\left[y^L \bar{X}^C + x^L \bar{Y}^C - x^L y^L\right]^L, \left[y^U \bar{X}^C + x^U \bar{Y}^C - x^U y^U\right]^L\right), \tag{3.58}$$
$$z^{cc} = \min\left(\left[y^L \bar{X}^C + x^U \bar{Y}^C - y^L x^U\right]^U, \left[y^U \bar{X}^C + x^L \bar{Y}^C - y^U x^L\right]^U\right). \tag{3.59}$$

and $\bar{\mathcal{X}} = \mathrm{Cut}(\mathcal{X})$ and $\bar{\mathcal{Y}} = \mathrm{Cut}(\mathcal{Y})$.

As shown in [170, Chapter 2], the definition above gives valid convex relaxations.

Lemma 3.9.19. $\times(\mathcal{X}, \mathcal{Y})$ *is (1,2)-convergent on* $\mathbb{M}X^0 \times \mathbb{M}Y^0$ *for any interval* $X^0 \times Y^0$.

Proof. Choose any $(\mathcal{X}, \mathcal{Y}) \in \mathbb{M}X^0 \times \mathbb{M}Y^0$. We have $w(\mathcal{X}\mathcal{Y}) \leq z^{cc} - z^{cv}$. There are four cases to consider. For the first case,

$$z^{cc} - z^{cv} = \left[y^L \bar{X}^C + x^U \bar{Y}^C - y^L x^U\right]^U - \left[y^L \bar{X}^C + x^L \bar{Y}^C - x^L y^L\right]^L. \tag{3.60}$$

Writing $r^U = w(R) + r^L$ for $R = \left[y^L \bar{X}^C + x^U \bar{Y}^C - y^L x^U\right]$ on the right,

$$z^{cc} - z^{cv} = w(\left[y^L \bar{X}^C + x^U \bar{Y}^C - y^L x^U\right]) + \left[y^L \bar{X}^C + x^U \bar{Y}^C - y^L x^U\right]^L - \left[y^L \bar{X}^C + x^L \bar{Y}^C - x^L y^L\right]^L, \tag{3.61}$$
$$= w(\left[y^L \bar{X}^C + x^U \bar{Y}^C\right]) + \left[y^L \bar{X}^C\right]^L + \left[x^U \bar{Y}^C\right]^L - y^L x^U - \left[y^L \bar{X}^C\right]^L - \left[x^L \bar{Y}^C\right]^L + x^L y^L, \tag{3.62}$$
$$\leq |y^L| w(\bar{X}^C) + |x^U| w(\bar{Y}^C) + \left[x^U \bar{Y}^C\right]^L - y^L x^U - \left[x^L \bar{Y}^C\right]^L + x^L y^L, \tag{3.63}$$
$$= |y^L| w(\mathcal{X}) + |x^U| w(\mathcal{Y}) + \left[x^U \bar{Y}^C - y^L x^U\right]^L - \left[x^L \bar{Y}^C - x^L y^L\right]^L, \tag{3.64}$$
$$= |y^L| w(\mathcal{X}) + |x^U| w(\mathcal{Y}) + \left[x^U (\bar{Y}^C - y^L)\right]^L - \left[x^L (\bar{Y}^C - y^L)\right]^L. \tag{3.65}$$

Noting that $\bar{Y}^C \subset Y^B$, it follows that every element of $\bar{Y}^C - y^L$ is nonnegative and bounded above by $w(Y^B)$. Thus, $\exists q_1, q_2 \in (\bar{Y}^C - y^L)$, both bounded betwen 0 and $w(Y^B)$, satisfying

$$
\begin{aligned}
z^{cc} - z^{cv} &\leq |y^L| w(\mathcal{X}) + |x^U| w(\mathcal{Y}) + x^U q_1 - x^L q_2, && (3.66)\\
&= |y^L| w(\mathcal{X}) + |x^U| w(\mathcal{Y}) + (x^L + w(X^B)) q_1 - x^L q_2, && (3.67)\\
&= |y^L| w(\mathcal{X}) + |x^U| w(\mathcal{Y}) + x^L (q_1 - q_2) + w(X^B) q_1, && (3.68)\\
&\leq |y^L| w(\mathcal{X}) + |x^U| w(\mathcal{Y}) + |x^L| w(\bar{Y}^C - y^L) + w(X^B) w(Y^B), && (3.69)\\
&\leq |y^L| w(\mathcal{X}) + (|x^U| + |x^L|) w(\mathcal{Y}) + w(X^B) w(Y^B). && (3.70)
\end{aligned}
$$

By similar arguments for the remaining three cases,

$$
w(\mathcal{X}\mathcal{Y}) \leq \tau_1 w(\mathcal{X} \times \mathcal{Y}) + \tau_2 w(X^B \times Y^B)^2, \tag{3.71}
$$

with $\tau_1 = 3 \max\{|x^L|, |x^U|, |y^L|, |y^U|\}$ and $\tau_2 = 1$. □

3.9.7.3 Univariate Functions

Assumption 3.9.20. *For every $(u, B, \mathbb{R}) \in \mathcal{L}$, functions $u^{cv}, u^{cc} : \bar{B} \to \mathbb{R}$, where $\bar{B} \equiv \{(X, x) \in \mathbb{I}B \times B : x \in X\}$, and $x^{\min}, x^{\max} : \mathbb{I}B \to \mathbb{R}$ are known such that*

1. *For every $X \in \mathbb{I}B$, $u^{cv}(X, \cdot)$ and $u^{cc}(X, \cdot)$ are convex and concave relaxations of u on X, respectively.*

2. *$x^{\min}(X)$ and $x^{\max}(X)$ are a minimum of $u^{cv}(X, \cdot)$ on X and a maximum of $u^{cc}(X, \cdot)$ on X, respectively.*

3. *For any $X_1, X_2 \in \mathbb{IR}$ with $X_2 \subset X_1$, $u^{cv}(X_1, x) \leq u^{cv}(X_2, x)$ and $u^{cc}(X_1, x) \geq u^{cc}(X_2, x)$ for all $x \in X_2$.*

4. *$u^{cv}([x, x], x) = u^{cc}([x, x], x)$ for every $x \in B$.*

In [170, §2.8], suitable $u^{cv}, u^{cc}, x^{\min}$, and $x^{\max}$ functions are given for the univariate functions:

- $x \mapsto x + c$, where $c \in \mathbb{R}$,

- $x \mapsto cx$, where $c \in \mathbb{R}_+$,
- $x \mapsto -x$,
- $x \mapsto \frac{1}{x}$,
- $x \mapsto \exp x$,
- $x \mapsto \ln x$,
- $x \mapsto x \ln x$,
- $x \mapsto \sqrt{x}$,
- $x \mapsto x^{2n}$, where $n = 1, 2, \ldots$,
- $x \mapsto x^{2n+1}$, where $n = 1, 2, \ldots$,
- $x \mapsto \sin x$, and
- $x \mapsto \cos x$.

Suitable functions can be readily derived for other univariate functions using the techniques of [125, §4].

McCormick's composition rule now defines relaxation functions for the elements of $\mathcal{L}$ as follows.

Definition 3.9.21. For every $(u, B, \mathbb{R}) \in \mathcal{L}$, define $(u, \mathbb{M}B, \mathbb{M}\mathbb{R})$ by

$$u(\mathcal{X}) = \big(u(X^B), \big[u^{cv}(X^B, \text{mid}(x^{cv}, x^{cc}, x^{\min}(X^B))), \\ u^{cc}(X^B, \text{mid}(x^{cv}, x^{cc}, x^{\max}(X^B)))\big]\big),$$

where $u(X^B)$ is the value of $(u, \mathbb{I}B, \mathbb{I}\mathbb{R})$ at X^B.

Note that $\mathcal{X} \in \mathbb{M}B$ implies that either $x^{cv} \in X^B$ or $x^{cc} \in X^B$, or both. By definition $x^{\min}(X^B), x^{\max}(X^B) \in X^B$, so that, in both uses of the mid function above, at least two of the three arguments lie in X^B. It follows that the mid function chooses an element of X^B, and hence of B, in both cases, so that $u(\mathcal{X})$ is well-defined.

Theorem 3.9.22 ([170, Theorem 2.4.27])**.** *$(u, \mathbb{M}B, \mathbb{M}\mathbb{R})$ is a McCormick extension of $(u, B, \mathbb{R})$.*

Lemma 3.9.23. *$(u, \mathbb{M}B, \mathbb{M}\mathbb{R})$ is (1,2)-convergent on $\mathbb{M}X^0$ for every interval $X^0 \subset B$.*

Proof. Choose any $\mathcal{X} \in \mathbb{M}X^0$. Since both $\mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B))$ and $\mathrm{mid}(x^{cv}, x^{cc}, x^{\max}(X^B))$ are in $X^C \cap X^B$, it follows that

$$|\mathrm{mid}(x^{cv}, x^{cc}, x^{\max}(X^B)) - \mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B))| \leq w(\mathcal{X}). \tag{3.72}$$

Now,

$$\begin{aligned}
w(u(\mathcal{X})) &= |u^{cc}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\max}(X^B))) - u^{cv}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B)))|, && (3.73)\\
&\leq |u^{cc}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\max}(X^B))) - u(\mathrm{mid}(x^{cv}, x^{cc}, x^{\max}(X^B)))| && (3.74)\\
&\quad + |u(\mathrm{mid}(x^{cv}, x^{cc}, x^{\max}(X^B))) - u(\mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B)))| && (3.75)\\
&\quad + |u(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B))) - u^{cv}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B)))|, && (3.76)\\
&\leq \tau w(X^B)^2 + L w(\mathcal{X}) + \tau w(X^B)^2, && (3.77)
\end{aligned}$$

where L is the Lipschitz constant for u on X^B. Thus, $\tau_1 = L$ and $\tau_2 = 2\tau$. □

An alternate proof could use a Lipschitz constant for either $u^{cv}(X^B, \cdot)$ or $u^{cc}(X^B, \cdot)$ and avoid using u altogether:

$$\begin{aligned}
w(u(\mathcal{X})) &= |u^{cc}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\max}(X^B))) - u^{cv}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B)))|, && (3.78)\\
&\leq |u^{cc}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\max}(X^B))) - u^{cc}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B)))| && (3.79)\\
&\quad + |u^{cc}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B))) - u^{cv}(X^B, \mathrm{mid}(x^{cv}, x^{cc}, x^{\min}(X^B)))|, && (3.80)\\
&\leq L w(\mathcal{X}) + \tau w(X^B)^2. && (3.81)
\end{aligned}$$

3.9.7.4 (1,2)-Convergence of natural McCormick extensions

Theorem 3.9.24. *Let $\mathbf{f} : D \to \mathbb{R}^n$ be an $\mathcal{L}$-factorable function with natural McCormick extension $\{\mathbf{f}\} : \mathcal{D} \subset \mathbb{M}D \to \mathbb{M}\mathbb{R}^n$. For any interval X^0 represented in $\mathcal{D}$, $\{\mathbf{f}\}$ is (1,2)-*

convergent on $\mathbb{M}X^0$; i.e., $\exists \tau_1, \tau_2 \in \mathbb{R}_+$ such that

$$w(\{\mathbf{f}\}(\mathcal{X})) \leq \tau_1 w(\mathcal{X}) + \tau_2 w(X^B)^2, \quad \forall \mathcal{X} \in \mathbb{M}X^0. \tag{3.82}$$

Proof. This follows immediately by repeated composition, addition, and multiplication. □

From here, we can recover all manner of more complicated-looking results. It follows that

$$w(F^C(\mathcal{X})) \leq \tau_1 w(X^C) + \tau_2 w(X^B)^2, \quad \forall \mathcal{X} \in \mathbb{M}X^0. \tag{3.83}$$

If $\mathcal{X} = (X^B, [\mathbf{x}, \mathbf{x}])$, we obtain

$$\begin{aligned} |\mathbf{f}^{cc}((X^B, [\mathbf{x}, \mathbf{x}])) - \mathbf{f}^{cv}((X^B, [\mathbf{x}, \mathbf{x}]))| &= w(F^C(\mathcal{X})) & (3.84) \\ &\leq \tau_2 w(X^B)^2, \quad \forall \mathbf{x} \in X^B, \quad \forall X^B \subset X^0. & (3.85) \end{aligned}$$

If $\mathcal{X}$ is obtained as the relaxation of an inner function; i.e., $\mathcal{X} = \mathcal{X}(\mathcal{P})$ where $\mathcal{X}$ is (1,2)-convergent on some $\mathbb{M}P^0$ and $\mathcal{P} = (P, [\mathbf{p}, \mathbf{p}]) \in \mathbb{M}P^0$, then we simply use the composition result to observe that $\{\mathbf{f}\} \circ \mathcal{X}$ is (1,2)-convergent on $\mathbb{M}P^0$, and hence

$$\begin{aligned} &|\mathbf{f}^{cc}((X^B(P), [\mathbf{x}^{cv}((P, [\mathbf{p}, \mathbf{p}])), \mathbf{x}^{cc}((P, [\mathbf{p}, \mathbf{p}]))])) - \mathbf{f}^{cv}((X^B(P), [\mathbf{x}^{cv}((P, [\mathbf{p}, \mathbf{p}])), \mathbf{x}^{cc}((P, [\mathbf{p}, \mathbf{p}]))]))| \\ &\qquad = w(F^C(\mathcal{X}(\mathcal{P}))) \leq \tau_1 w([\mathbf{p}, \mathbf{p}]) + \tau_2 w(P)^2 = \tau_2 w(P)^2, \quad \forall \mathbf{p} \in P, \quad \forall P \subset P^0, \end{aligned}$$

where τ_2 is only a function of P^0 and is a combination of the convergence constants for $\{\mathbf{f}\}$ and $\mathcal{X}$.

3.10 Supplementary material

When solving the lower-bounding problem in a global minimization problem, it is sometimes more efficient to minimize a linearization of the convex underestimating objective rather than recomputing the nonlinear relaxation at every step in the numerical optimizer. This is because the linearizations are often not much weaker than the nonlinear relaxation, yet they are a great deal cheaper to calculate for any nontrivial dynamic optimization problem. The following proposition gives a bound on the distance between a relaxation and its linearization at a point $\mathbf{p}^*$ that is contained in a neighborhood where the relaxation is twice continuously differentiable.

Proposition 3.10.1. *Let $P \subset \mathbb{R}^{n_p}$ and $\mathbf{f} : P \to X \subset \mathbb{R}^{n_x}$. Let $F^C : \mathbb{M}P \to \mathbb{IR}^{n_x}$ be a relaxation function for $\mathbf{f}$ in P that converges pointwise in P with order γ. Suppose there exists $\mathbf{p}^* \in \widehat{P}$ and $\epsilon > 0$ such that $\mathbf{f}^{cv}((\widehat{P}, \cdot))$ and $\mathbf{f}^{cc}((\widehat{P}, \cdot))$ are twice continuously differentiable on $N_\varepsilon(\mathbf{p}^*) \equiv \{\mathbf{p} \in \widehat{P} : \|\mathbf{p} - \mathbf{p}^*\| < \varepsilon\}$. Then the affine functions constructed at $\mathbf{p}^*$,*

$$\mathbf{a}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}}) = \mathbf{f}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}^*}) + (\nabla \mathbf{f}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}^*}))^{\mathrm{T}}(\mathbf{p} - \mathbf{p}^*), \quad \forall \mathbf{p} \in \widehat{P},$$

where $\widehat{\mathcal{P}}_{\mathbf{p}^*} \equiv (\widehat{P}, [\mathbf{p}^*, \mathbf{p}^*])$, make $(\mathbf{a}^{cv}, \mathbf{a}^{cc})$ a relaxation function in P that converges pointwise in $N_\varepsilon(\mathbf{p}^*)$ with order $\min\{\gamma, 2\}$.

Proof. The linearization of $\mathbf{f}^{cv}(\widehat{\mathcal{P}}_{\mathbf{p}})$ ($\mathbf{f}^{cc}((\widehat{P}, \cdot))$) is guaranteed to underestimate (overestimate) $\mathbf{f}$ due to convexity (concavity). Since $\mathbf{f}^{cv/cc}((\widehat{P}, \cdot))$ is twice differentiable in $N_\varepsilon(\mathbf{p}^*)$, there are mappings $\mathbf{H}^{cv/cc} : N_\varepsilon(\mathbf{p}^*) \to \mathbb{R}^{n_p \times n_p}$ such that

$$\begin{aligned}\mathbf{f}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}}) = \mathbf{f}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}^*}) &+ (\nabla \mathbf{f}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}^*}))^{\mathrm{T}}(\mathbf{p} - \mathbf{p}^*) \\ &+ \frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^{\mathrm{T}}\mathbf{H}^{cv/cc}(\mathbf{p})(\mathbf{p} - \mathbf{p}^*) + \mathbf{r}(\mathbf{p} - \mathbf{p}^*), \quad \forall \mathbf{p} \in N_\varepsilon(\mathbf{p}^*),\end{aligned}$$

where $\mathbf{r}$ satisfies

$$\lim_{\mathbf{p} \to \mathbf{p}^*} \frac{\|\mathbf{r}(\mathbf{p} - \mathbf{p}^*)\|}{\|\mathbf{p} - \mathbf{p}^*\|^2} = 0.$$

Then,

$$\mathbf{f}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}}) - \mathbf{a}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}}) = \frac{1}{2}(\mathbf{p} - \mathbf{p}^*)^{\mathrm{T}}\mathbf{H}^{cv/cc}(\mathbf{p})(\mathbf{p} - \mathbf{p}^*) + \mathbf{r}(\mathbf{p} - \mathbf{p}^*), \qquad \forall \mathbf{p} \in N_\varepsilon(\mathbf{p}^*),$$

so there exists $\tau > 0$ such that

$$\left|\mathbf{f}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}}) - \mathbf{a}^{cv/cc}(\widehat{\mathcal{P}}_{\mathbf{p}})\right| \leq \tau w\left(\widehat{P}\right)^2, \quad \forall \mathbf{p} \in N_{\varepsilon}(\mathbf{p}^*).$$

The result follows by applying the assumed pointwise convergence of order γ of F^C, using the triangle inequality, and taking the sup over $N_{\varepsilon}(\mathbf{p}^*)$. □

Remark 3.10.2. Numerically, we have assessed empirical Hausdorff convergence by constructing a nested sequence of intervals $\widehat{P}$ containing the global minimum for several test problems. For both ODE relaxation methods studied in this chapter (relaxation-amplifying dynamics and relaxation-preserving dynamics), the empirical Hausdorff convergence behavior for the linearized relaxations in the vicinity of the global minimum closely tracks that for the nonlinear relaxations for all test problems. We have not seen a case where the nonlinear relaxations produce drastically different convergence behavior from the linearized relaxations. The convergence order is about the same between nonlinear and linearized relaxations, but the linearized relaxations can have a slightly larger prefactor.

Chapter 4

Design, execution, and analysis of time-varying experiments for model discrimination and parameter estimation in microreactors

Abstract

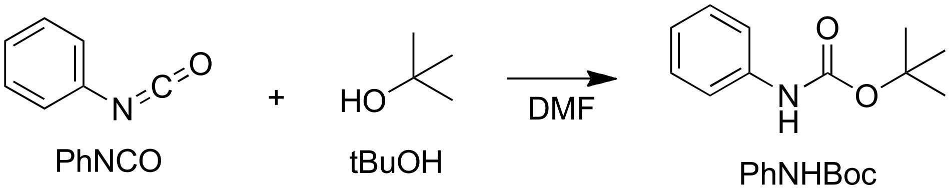
Time-varying, or dynamic, experiments can produce richer data sets than a sequence of steady-state experiments using less material and time. A case study demonstrating this concept for microreactor experiments is presented. Beginning with five kinetic model candidates for the reaction of phenylisocyanate with *t*-butanol, an initial dynamic experiment showed that two of the five models gave a similar quality of fit to the experimental data, whereas the remaining three gave significantly poorer fits. Next an optimal experiment was designed to discriminate between the remaining two models. This drove the two models to differ significantly in quality, leaving a single model and a set of kinetic parameter values that adequately described the data. This method can be applied to future kinetic studies to reduce material use and experimental time while validating a dynamic model of the physics and chemical kinetics.

4.1 Introduction

Most organizations continually strive to reduce costs and environmental impact while maintaining or improving products and processes. One way to achieve this aim is to reduce material and time used in experiments in research and development by creating more information-rich experiments through time-varying, or *dynamic* experiments. Dynamic experiments allow rapid exploration of the permissible experimental space without waiting for steady state between changes. *Model-based design of experiments* (MBDoE), or *optimal experimental design* (OED) has been applied to dynamic experiments for some time, but has not been used to design microreactor experiments. For a review of model-based design of experiments, see [70]. Such techniques use computer simulations that take the experimental conditions as input to estimate *a priori* the information content of an experiment. Numerical optimization software embeds the aforementioned computer simulation to maximize the expected information content. Using different objective functions, experiments can be designed to drive apart the predictions of two or more candidate models (*model discrimination* [34, 35, 41–43, 82, 88]) or to minimize the expected size of the confidence region for the model parameters [36, 70, 211]. Time-varying experiments are also useful because they allow validation of the dynamics of a model, which is especially useful when a dynamic model of the process will be used to design the control system for the process as in [105].

Due to small holdup, excellent heat transfer per unit volume, and lack of head space, microreactors and small-diameter tubular reactors allow conditions that could be dangerous in batch, such as those involving diazomethane [124], diazotization [220], nitration [50, 104, 219], high-pressure hydrogenation [203], and lithium-halogen exchange [39, 138, 139]. To our knowledge, MBDoE techniques have not yet been applied to designing time-varying experiments in microreactors. The combination of small holdup in the microreactor with a dynamic model allows relatively rapid changes in reaction conditions while extracting meaningful information from the process.

We studied the reaction of phenylisocyanate (PhNCO) with t-butanol (tBuOH) in *N,N*-dimethylformamide (DMF) to produce N-Boc-aniline (PhNHBoc) (Scheme 4.1). Isocyanates



Scheme 4.1: The principal reaction studied in this chapter

can be produced in a Curtius rearrangement of acyl azides which in turn can be produced by reaction of carboxylic acids with diphenylphosphoryl azide (DPPA) under basic conditions [182, 183]. Similarly to Scheme 4.1, various isocyanates can be reacted with various nucleophiles to produce ureas and carbamates. The former substructure is frequently found in enzyme inhibitors and pseudopeptides [122]. The reactivity of isocyanates with nucleophiles has been widely studied due to interest in production of urethanes and polyurethane [9–13], alcohol content measurement [145–147, 179, 180, 193], and more recently for analysis of environmental micro-pollutants [205].

Bauer et al. [15] and Galvanin et al. [72] did purely computational studies on the application of optimal experimental design techniques to the reaction of PhNCO with butanol (undisclosed isomer) in a semibatch reactor. In the present work, actual experiments were performed rather than a purely computational study. McMullen and Jensen [128] applied optimal experimental design techniques to microreactors to design a sequence of steady-state set points at which to measure reactor performance. In contrast, we consider time-varying input functions and continuous monitoring of concentration. Mozharov et al. [137] used step functions to interrogate an experimental system. We take this idea further by allowing arbitrary input functions rather than only step functions. Moore and Jensen [132, 133] used linear ramp functions in flow rate to emulate batch reactor time-course responses. The present method can be used with arbitrary time-varying input functions such as piecewise constant, piecewise linear, or sums of basis functions (e.g., flow rates, temperatures). In practice, the difficulty is that solving for the optimal experiment is more challenging in the present case since a broader range of possible experiments are considered.

A model for the reaction of butanol (undisclosed isomer) with phenylisocyanate was jointly developed by researchers at BASF and Universität Heidelberg [15, 101, 102]. They indicated mass-action kinetics for PhNCO and butanol coupling to form the carbamate product as well as for the reversible addition of PhNCO to the carbamate product to form an allophanate. For trimerization of PhNCO to form triphenyl isocyanurate, they indicated second-order kinetics.

Dyer and coworkers [63] reported that the reaction of isocyanate with *n*- and *s*-butyl alcohols in xylene has approximately second-order kinetics, with the value of the rate constant slightly larger for large excesses of alcohol. Activation energies were reported as 8.1 and 9.9 kcal/mol for the *n*- and *s*- isomers, respectively. Bailey and coworkers [8] reported the following relative activities of substituted phenylisocyanates toward alcohols: *m*-chlorophenylisocyanate > phenylisocyanate > *p*-tolyl isocyanate > *o*-tolyl isocyanate. Baker et al. [9, 11, 13] proposed that alcohol and isocyanate first form a complex which then reacts with a second alcohol molecule to form the product and release an alcohol molecule. Zaplatin et al. [222] studied the reaction of phenyl isocyanate with *n*-butanol in amides and dimethyl sulfoxide and proposed that the reaction is faster in these solvents because an alcohol-solvent complex is formed. Chang and Chen [49] showed that, considering a second-order rate law (first order in both phenyl isocyanate and isobutanol), the observed rate constant increases with the ratio $[\text{alcohol}]_0/[\text{isocyanate}]_0$. Plotting the data from Table IV of [49], it appears that k_{obs} is approximately constant until $[\text{alcohol}] \approx 0.8[\text{phenylisocyanate}]$ and for all higher concentrations of alcohol, there is a linear trend of increasing k_{obs}. Side products of the reactions of isocyanates with nucleophiles are given in Figure 6 of [216], and include uretdiones, isocyanurates, allophanates, and biurets. In a batch reaction between an isocyanate and an alcohol described in [169], carbamate was the primary product and allophanate was the next most abundant product. Schwetlick and Noack [169] claimed that the isocyanurate cyclotrimer is built up via a linear trimer adduct.

4.1.1 Overview of optimal experimental design procedure

Following is the sequence of steps we used to design and execute optimal dynamic experiments and discriminating between candidate models.

1. Gather system model(s) to be compared.

2. Implement the system model(s) in a dynamic simulation environment such as an ordinary differential equation (ODE) or differential-algebraic equation (DAE) simulator.

3. Select initial guesses for model parameters for each candidate model. For example, these initial guesses for the parameters may come from intuition, molecular modeling, or literature on similar systems. If insufficient information is available to select initial guesses for model parameters, design a simple experiment that explores the experimental space by varying flowrates, temperatures, etc. throughout their permissible ranges at permissible rates of change (e.g., temperature cannot be changed instantaneously). Our initial experimental conditions are depicted in Figure 4-1. Estimate parameters for each model using these data to minimize χ^2.

4. Design experiments to distinguish between models by maximizing their predicted deviations from one another; perform experiments; estimate parameters and calculate optimal χ^2 values.

5. Reject models with excessive lack-of-fit (e.g., based on χ^2 test).

6. If no models remain, select/generate additional models and go to step 2.

7. If two or more models remain, update the model parameters to the new fits from step 4, then begin the process again at step 4.

4.2 Experimental and computational methods

To perform the dynamic experiments, we used three syringe pumps (Harvard Apparatus PHD2000) to feed PhNCO and *t*BuOH solutions in DMF as well as neat DMF, into a silicon microreactor. For the initial experiment, concentrations of PhNCO and *t*BuOH were both 1.0 M; for the optimized experiment, both were 2.0 M. An FTIR flow cell (Mettler Toledo FlowIR) was used to estimate concentrations and National Instruments LabVIEW was used to control the syringe pump flow rates, send the temperature set point to the temperature controller, and write IR absorption data to the master output file. See Figure 4-2. At startup, to purge the system of any gas bubbles rapidly and without using any expensive

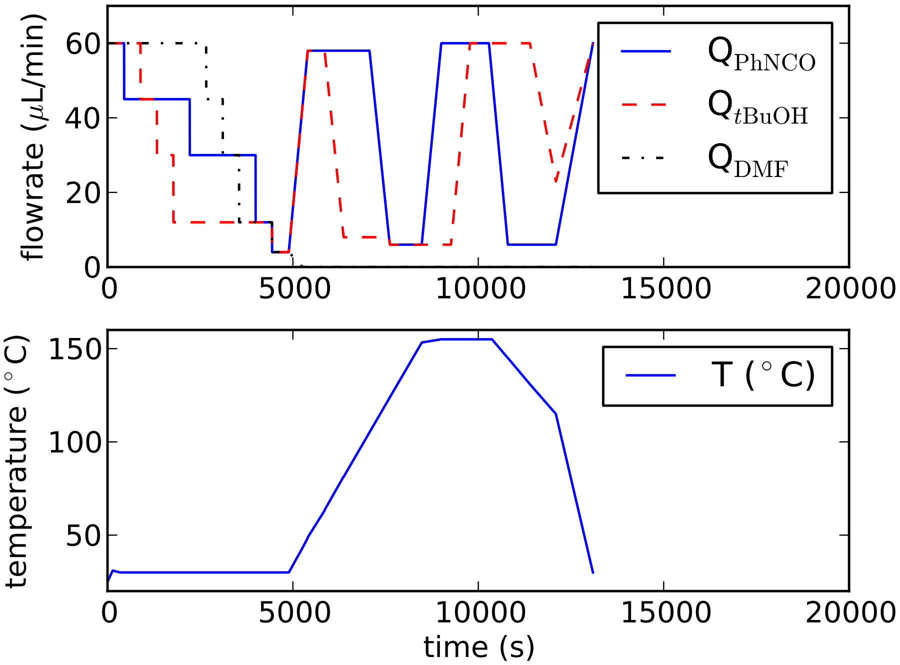


Figure 4-1: Experimental conditions for manually-designed initial experiment used to estimate parameters for all candidate models.

starting materials, the system was first flushed with neat solvent at 120 μL/min until no gas bubbles were visible in the microreactor or tubing.

The microfabricated silicon microreactor had the specifications described by McMullen and Jensen [128]. In particular, a reaction volume of 120 μL consisting of a channel with 0.4 mm×0.4 mm square cross section and a quench volume of 14.25 μL [153].

IR calibration curves were made for PhNCO and PhNHBoc using at least six samples with concentrations from 0 to 1.0 M with at least two measurements per sample. Note: PhNCO is a lachrymator and should be handled in a fume hood. We made PhNHBoc for IR calibration: to a round-bottom flask were added 3 g of PhNCO, 2.5 eq of *t*BuOH and 18 g of toluene. The mixture was heated to reflux for 4 h and concentrated to a crystalline solid powder under vacuum. Yield of crude product was 94%. ^{1}H-NMR was used to confirm product identity as PhNHBoc (§4.6.2). Crude product was used for FTIR calibration curve.

To estimate concentration data from IR data, we used the ranges 2350–2285 cm^{-1} for PhNCO and 1166–1162 and 1319–1315 cm^{-1} for PhNHBoc. These ranges were found to give the most accurate calibrations among wavenumbers for which all other known species

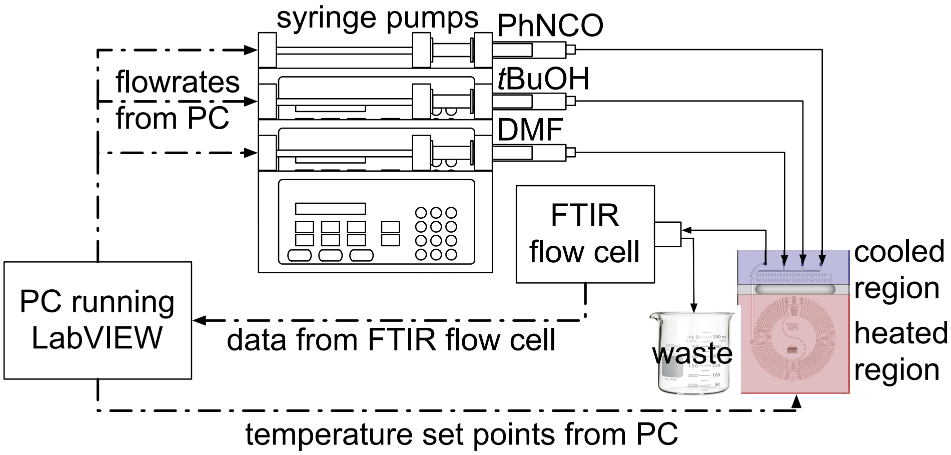


Figure 4-2: Experimental apparatus used a PC to control inlet flow rates of reactants, inlet flow rate of neat solvent, and temperature of microreactor while collecting IR data. Solid lines show material flow; dot-dashed lines show information flow.

in the system absorbed weakly. Although this led to larger mean squared error *in the calibration curves* than the chemometric techniques described in [202], it produced dramatically smoother *time-series data* than the chemometric techniques. This is because chemometric techniques work best when using a training data set with samples containing known concentrations of all species that will occur in the experimental mixture, whereas we gathered data from one species in solvent at a time.

We considered five candidate models for the reaction of PhNCO with *t*BuOH (Table 4.1).

Table 4.1: Five kinetic models were considered. In all cases, we used the Arrhenius temperature-dependence $k = k_0 \exp(-E_a/(RT))$ and the free parameters k_0 and E_a.

name	rate expression
m01	$kC^0_{\mathrm{PhNCO}}C^1_{t\mathrm{BuOH}}$
m10	$kC^1_{\mathrm{PhNCO}}C^0_{t\mathrm{BuOH}}$
m11	$kC^1_{\mathrm{PhNCO}}C^1_{t\mathrm{BuOH}}$
m12	$kC^1_{\mathrm{PhNCO}}C^2_{t\mathrm{BuOH}}$
m21	$kC^2_{\mathrm{PhNCO}}C^1_{t\mathrm{BuOH}}$

4.2.1 Dynamic model for time-varying experiments

For design of time-varying experiments, as well as subsequent analysis of experimental data to discriminate between models and estimate parameter values, a DAE model was simulated in the process simulator Jacobian (RES Group, Inc.). The nonlinear optimizer initiates a

Jacobian simulation whose output is used to calculate the objective function (χ^2 for parameter estimation, divergence between models as in [43] for designing model discrimination experiments, or determinant of the Fisher information matrix [35, 210] for designing parameter estimation experiments). The decision variables in the nonlinear optimizer vary depending on the problem currently being solved. For parameter estimation problems, the decision variables are the model parameters whereas for experimental design problems, the decision variables are the control parameters.

Jacobian was selected for the simulation for two primary reasons. First, it exploits the sparsity of the dynamic system—the fact that there are of hundreds or thousands of state variables and equations but most variables only appear in a handful of the equations—to simulate the system very efficiently. Second, it allowed separating the physical models for the microreactor, quench, and IR flow cell from the models for the chemistry. This minimized the amount of redundant lines of modeling code to avoid repetition for each new sequence of conditions when a new experiment is performed and whenever changes are made to the experimental apparatus.

A dynamic model for the microreactor was derived, assuming incompressible solutions and cross-sectionally uniform concentration, yielding the following partial differential equation (PDE) in time and the axial spatial dimension:

$$\frac{\partial C_i}{\partial t} + v_x \frac{\partial C_i}{\partial x} = D_i \frac{\partial^2 C_i}{\partial x^2} + R_i, \quad i = 1, \ldots, n_{\text{species}}.$$

The Peclet number (Pe) is very large for our system (about 10^5), indicating that transport is dominated by convection rather than diffusion. Since centered differences can yield unphysical oscillations for large values of Pe, upwind differences [16] were used for the first-order derivatives in the finite-volume model. We used 30 finite volumes for the reaction portion and 10 finite volumes for the quench portion and tubing between quench and IR flow cell. Discretizing the PDE yields the ordinary differential equation (ODE):

$$\frac{dC_{i,j}}{dt} = -v_x \frac{C_{i,j} - C_{i,j-1}}{\Delta x} + D_i \frac{C_{i,j-1} + C_{i,j+1} - 2C_{i,j}}{(\Delta x)^2} + R_{ij},$$

$$i = 1, \ldots, n_{\text{species}}, \quad j = 1, \ldots, n_{\text{mesh}}.$$

Upwind differences can yield excessive “numerical diffusion” as an artifact. To check for this phenomenon, we used step functions in the input flow rates of reactants. The shapes of the time-course experimental concentration measurements agreed closely with those of the simulation (Figure 4-3), validating our finite-volume model. The step functions also helped ensure good synchronization of IR measurement data to the experimental conditions. Simulations reported for this 120 μL microreactor in the supporting information of [128] indicated that, under the conditions reported there, the reactor would be adequately modeled by plug flow for residence times of 1–10 min and temperatures of 50–150°C. We expected the Boc group formed in the coupling reaction to decompose at significant rates at about 150°C [3] and appreciable rates as low as 130°C [213], which was not accounted for in our kinetic model. Therefore, we kept the reactor temperature below 130°C after completing the initial experiment.

The gradient-based optimization solver SNOPT [74] was used to optimize the model, which was simulated using Jacobian. SNOPT has often been favored for optimal control problems because it only requires first derivatives and it typically requires relatively few objective function evaluations. Since the objective function evaluation requires a dynamic simulation, it dominates the computational cost of the optimization and fewer objective function evaluations implies less CPU time. The optimization was repeated from at least 20 random initial guesses for the model parameters since problems of this type tend to be nonconvex and have suboptimal local minima [188].

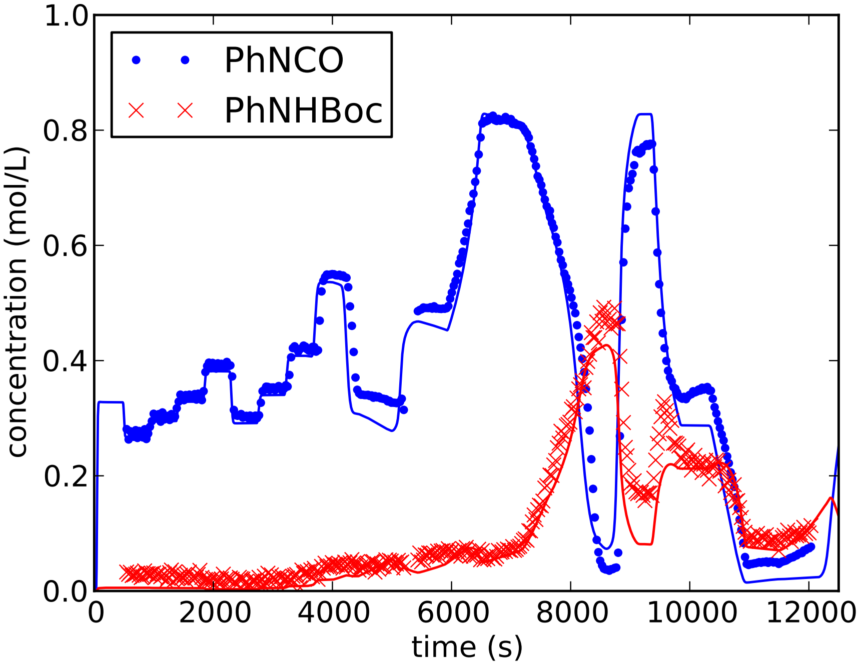
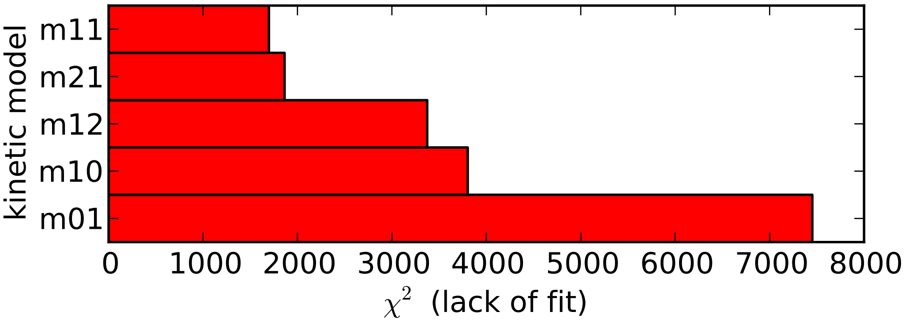


Figure 4-3: Plot of time-series data for manually-designed experiment (Figure 4-1) and best-fitting dynamic model, m11. Points show experimental data from IR; curves show model fit. The first 3500 seconds of data show that the amount of dispersion in the dynamic model closely approximates the dispersion in the experimental data since there is a similar level of smoothing of step functions of PhNCO concentration in model and experiment. This experiment used about 7 mmol of PhNCO and 6 mmol of *t*BuOH.

4.3 Results and discussion

4.3.1 Lack of fit for each model considered

After performing the initial experiment with time-varying conditions given in Figure 4-1, we fit the parameters to the data for each model in turn by minimizing χ^2 in the dynamic simulation. The best-fit parameter and χ^2 values for each kinetic model (Table 4.1) are shown in Figure 4-4. The best fits for models m11 and m21 were of similar quality, whereas those for the remaining models were significantly poorer. See the Figure 4-3 for the fit of the best model, m11, to the data from the initial experiment. Having eliminated all models except m11 and m21 from consideration, we designed an experiment to discriminate between those two following the methods of Buzzi-Ferraris and Manenti [43]. Essentially, we used the best-fit model parameters obtained from the initial experiment and maximized the differences in predicted measurements (weighted by their uncertainties) for the two models. The designed experimental conditions are shown in Figure 4-5.



model	k_0 ($\mathrm{M}^{-1}\mathrm{s}^{-1}$)	E_a (kJ/mol)
m11	7.15×10^1	28.0
m21	1.28×10^3	34.2
m12	4.27×10^2	29.5
m10	2.37×10^1	31.5
m01	5.85×10^6	73.3

Figure 4-4: Models m11 and m21 show similar lack of fit; remaining three models give substantially worse fits, with 2 to 4.4 times more error than the best model, and can be eliminated from further experimentation.

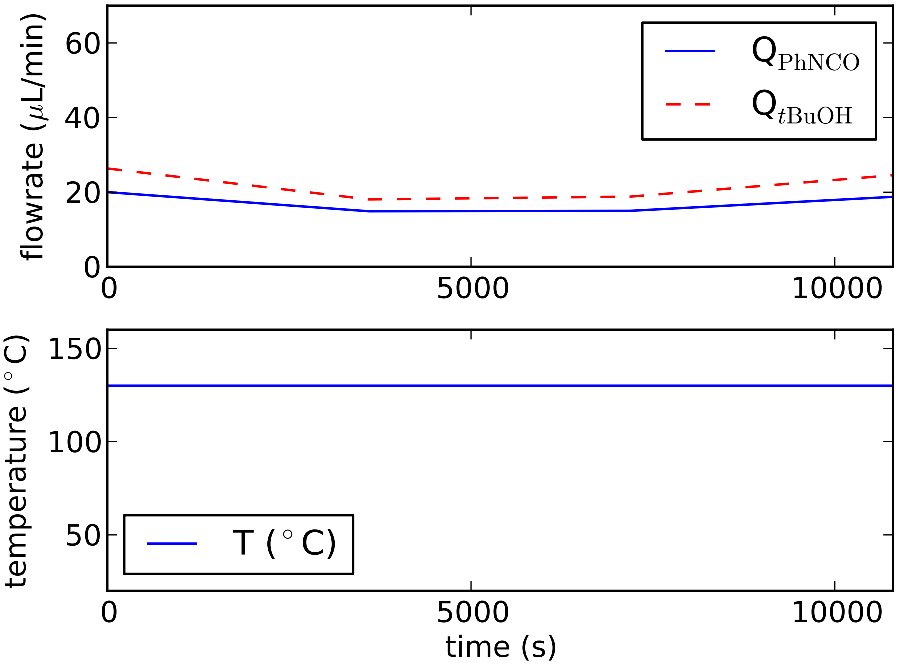


Figure 4-5: Experimental conditions for optimal dynamic experiment to discriminate between models m11 and m21.

4.3.2 Results of model discrimination experiment

After designing and executing an experiment to discriminate between the remaining candidate models, m11 and m21, we plotted the simulated trajectories of PhNCO and PhNHBoc concentration using the parameter values found using only the initial experiment. The discriminating experiment succeeded in driving the concentration trajectories apart (Figure 4-6).

Next, we used all data from the initial and model discrimination experiments simultaneously to estimate the best-fit parameters for models m11 and m21. Although this improved the fit of model m21 to the new experimental data significantly, it became clear that model m11 was significantly better at predicting all of the experimental data for the two different experiments, since model m11 had $\chi^2 = 2115.4$ whereas model m21 had $\chi^2 = 2441.3$. See Figure 4-7. The values of χ^2 stated were calculated assuming that the only source of error in the experiment is the IR measurement. Although this indicates both models are likely to be inadequate based on the χ^2 tests, it also indicates that model m11 is 10^{26} times more

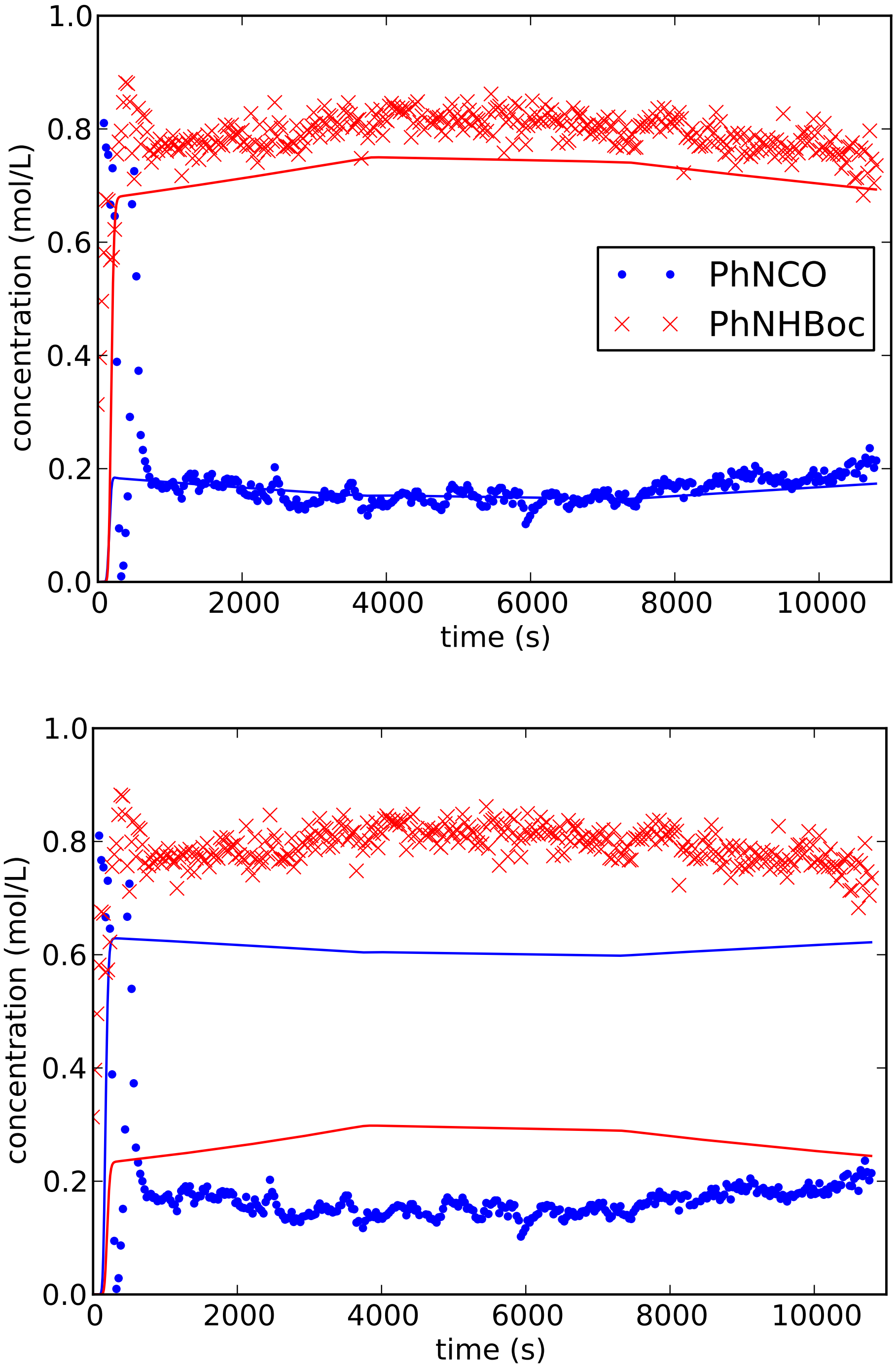


Figure 4-6: Simulated trajectories and experimental data for model discrimination experiment using best-fit parameter values from initial experiment only. Top: model m11, $\chi^2 = 652.3$; bottom: model m21, $\chi^2 = 65101.6$. This experiment used about 10 mmol each of PhNCO and tBuOH.

probable than model m21.

4.3.3 Reasons for imperfect fit

There are a few reasons for imperfect fits in the dynamic model. First, there may be reactions occurring that are not present in our model, such as Boc deprotection, which occurs at significant rates at temperatures around 130°C or higher [3, 213], and the formation of side products. Second, the dynamics of heating of the silicon microreactor via the aluminum chuck are not modeled. The temperature of the microreactor is assumed to be equal to the set point at all times. We chose temperature ramp rates sufficiently small (between -2°C/min and $+3$°C/min) that temperature of the microreactor followed the set point within ±2°C. McMullen [127, Chapter 4] also showed that with proper tuning of the temperature controller, the temperature of the microreactor closely follows the set point. Third, syringe pumps are known to be imperfect in their delivery of material. In some cases, the flowrates delivered by multiple syringe pumps have been observed to oscillate; we mitigated this effect by used a 250-psi backpressure regulator to give a relatively constant resistance to the syringe pumps, with the added benefit of preventing *t*BuOH from boiling at the elevated temperatures used in our experiments.

4.3.4 Prediction accuracy for reactor performance at steady state

To validate the model discrimination and parameter estimation techniques used, the system was run to steady state at selected residence times and temperatures out of the permissible ranges of 1–10 minutes and 50–130°C. Model m11 with the best-fit parameters from the combined experiment was used to calculate the theoretical concentrations of phenylisocyanate and N-Boc-aniline at steady state. The root mean square (RMS) errors for PhNCO and PhNHBoc predictions were 0.071 M and 0.078 M, respectively. See Figure 4-8.

4.4 Conclusion

We used a microreactor system with online FTIR spectroscopy to execute optimal dynamic experiments to validate a dynamic model of the microreactor system, distinguish between

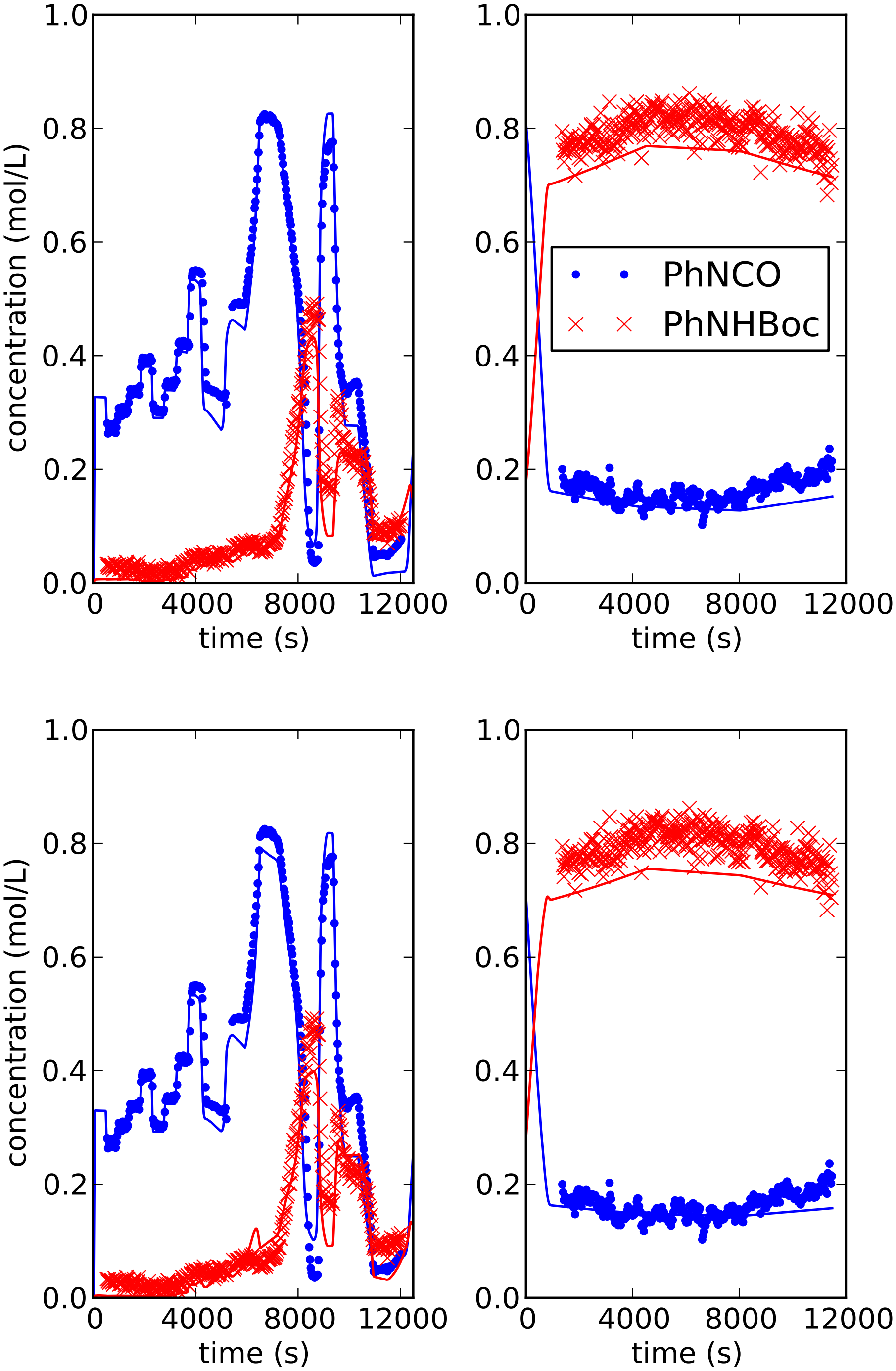
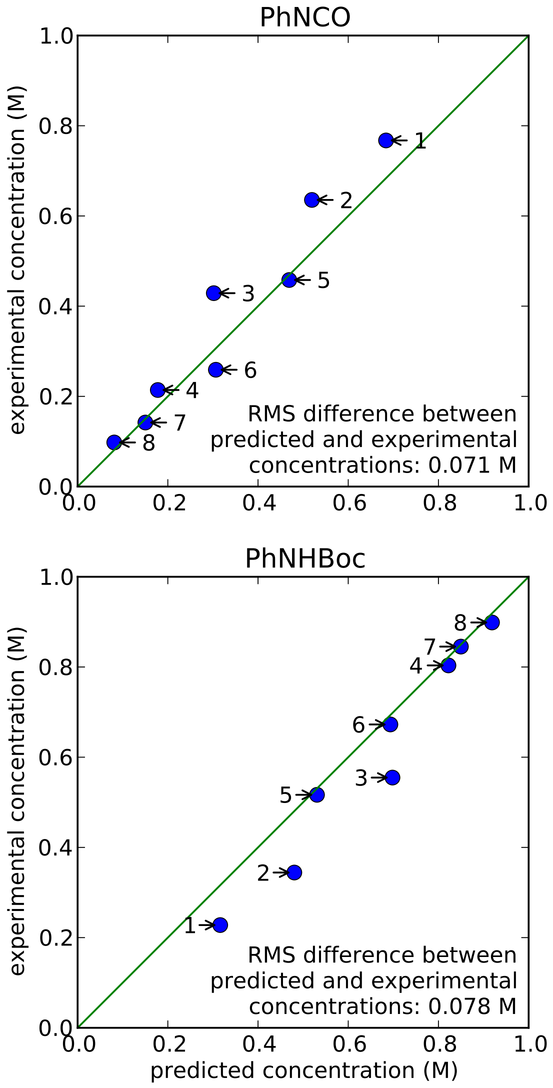


Figure 4-7: Simulated trajectories and experimental data for both initial experiment and optimal experiment for model discrimination using best-fit parameter values obtained using all experimental data. Top: model m11, $\chi^2 = 2115.4$, $k_0 = 63$ M^{-1} s^{-1}, $E_a = 27$ kJ/mol; bottom: model m21, $\chi^2 = 2441.3$, $k_0 = 1400$ M^{-2} s^{-1}, $E_a = 33$ kJ/mol. Points show experimental data; curves show simulated concentrations.



label	temperature (°C)	residence time (min)
1	90	1
2	90	2
3	90	5
4	90	10
5	130	1
6	130	2
7	130	5
8	130	10

Figure 4-8: Parity plot for predictions from dynamic model m11 and experimental measurements at steady state with annotations for residence times and temperatures. The steady-state experiments used about 10 mmol each of PhNCO and *t*BuOH. The RMS difference between the experimental and predicted PhNCO concentrations is 0.071 M. That for PhNHBoc is 0.078 M.

candidate chemical kinetic models, and estimate best-fit kinetic parameters. Such experiments have the potential to reduce experimental time and material used, with commensurate reduction in cost, by obtaining information at a higher rate per unit of time and material. In particular, the initial time-varying experiment used about 7 mmol of PhNCO and 6 mmol of *t*BuOH over 3.5 hours and provided useful data at 377 time points and allowed rejecting three of the five models whereas the steady-state experiments used about 10 mmol each of PhNCO and *t*BuOH over 2.3 hours and provided data at 8 different experimental conditions. Applying the χ^2 test to the steady-state experimental data is insufficient to reject any of the five models. Furthermore, the steady-state experiments, having used fixed feed concentrations of PhNCO and *t*BuOH both equal to 2.0 M, cannot distinguish model m21 from m12 nor model m10 from m01.

Going forward, we recommend microreactor experiments with two or three initial step functions in concentration to verify proper synchronization between experimental conditions and IR data measurement as well as adequate modeling of the fast dynamics of the physical system. Such step functions could be followed by further step or ramp functions for programming simplicity, or by control functions discretized using orthogonal polynomials such that the control profiles are smooth, making the DAE simulation more CPU-efficient. On-line HPLC as used in previous studies [128] would also be helpful to determine online the number of species of significant concentration, to see whether the model captures the correct number of species and to validate the concentrations estimated using IR.

4.5 Acknowledgments

Thanks are due to John E. Tolsma (RES Group, Inc.) for support with interfacing Jacobian with the external optimizer and Brandon J. Reizman and Stephen C. Born for assistance with LabView software and experimental setup. Novartis Pharmaceuticals, Inc. is gratefully acknowledged for financial support.

4.6 Supporting information

4.6.1 Experimental and computational details

The effective diffusion coefficient including numerical diffusion from discretization with upwind differences is given by [87]

$$D_{\text{eff}} = D + \frac{v\Delta z}{2},$$

where D is the true physical diffusion coefficient, v is the superficial velocity of the fluid, and Δz is the length of a finite control volume used in the discretization. For our system, the values are:

$$\begin{aligned}
\Delta z &= \frac{V_{\text{rctr}}}{n_{\text{mesh}} A_{\text{XS}}} = \frac{120 \text{ mm}^3}{30 \cdot 0.4 \text{ mm} \cdot 0.4 \text{ mm}} = 25 \text{ mm}, \\
v_{\max} &= \frac{Q_{\max}}{A_{\text{XS}}} = \frac{120 \text{ mm}^3/\text{min}}{0.4 \text{ mm} \cdot 0.4 \text{ mm}} = 750 \text{ mm/min} = 12.5 \text{ mm/s}, \\
D &= 2{\times}10^{-3} \text{ mm}^2/\text{s}, \\
D_{\text{eff,max}} &= D + \frac{v_{\max}\Delta z}{2} = 2{\times}10^{-3} \text{ mm}^2/\text{s} + \frac{12.5 \text{ mm/s} \cdot 25 \text{ mm}}{2} = 2{\times}10^{-3} \text{ mm}^2/\text{s} + 156.25 \text{ mm}^2/\text{s}, \\
&= 7.8126{\times}10^4 D.
\end{aligned}$$

Although the effective diffusion coefficient is five orders of magnitude larger than the molecular diffusion coefficient, we still obtained accurate modeling of the smoothing of step functions in concentration input to the system. In the physical system, we expect the effective diffusivity, or *dispersivity*, to be larger than the molecular diffusivity due to the nonuniform velocity profile in the channel of the reactor. Deen [60, §9.7] gives the following formula for dispersivity, K, for flow in a tube, which in turn is based on [6, 196]:

$$K = D\left(1 + \frac{\text{Pe}^2}{192}\right) = D + \frac{vr^2}{48D}.$$

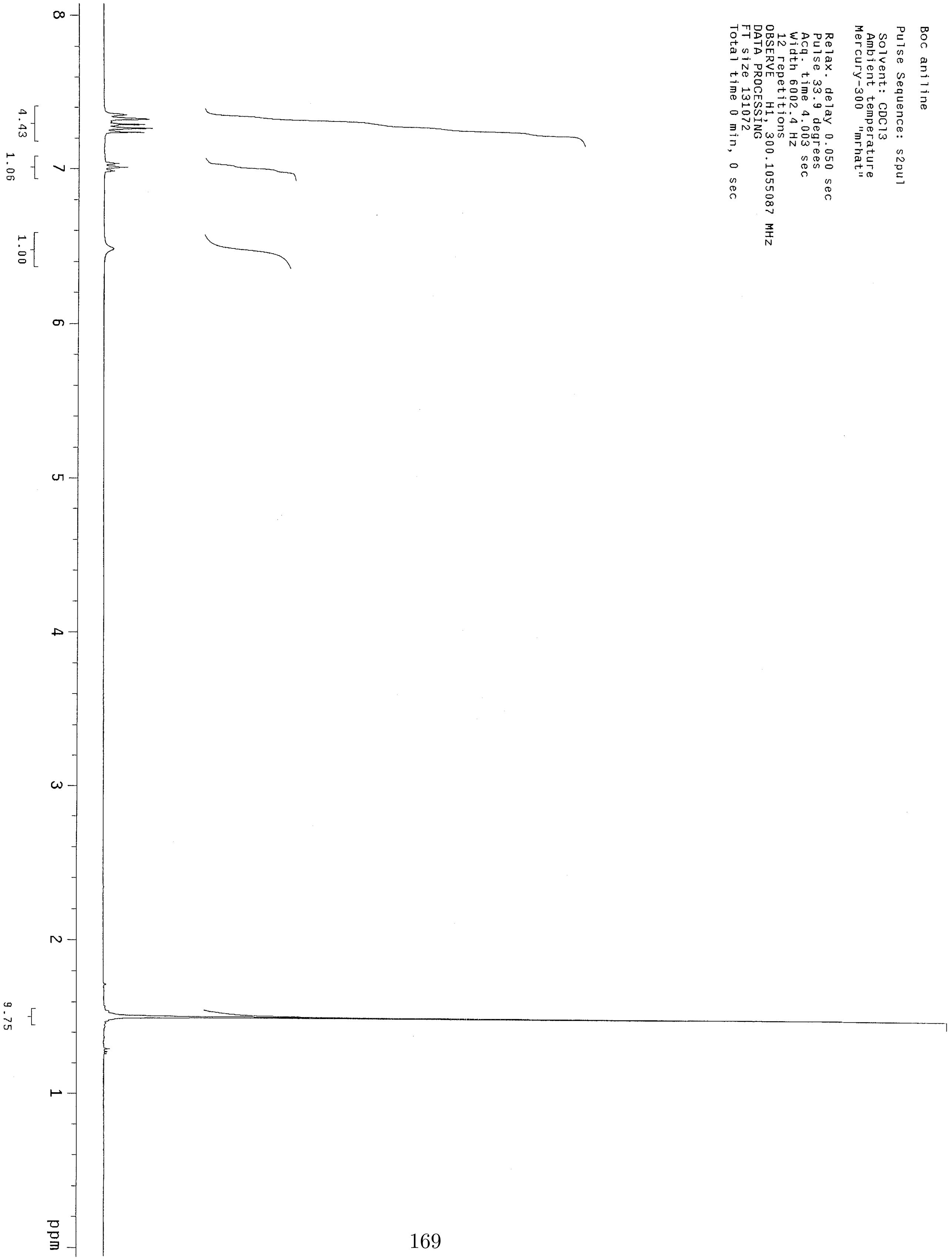
The square channel in the reactor is 0.4 mm×0.4 mm. If we take use 0.2 mm for the radius r in the formula, we obtain:

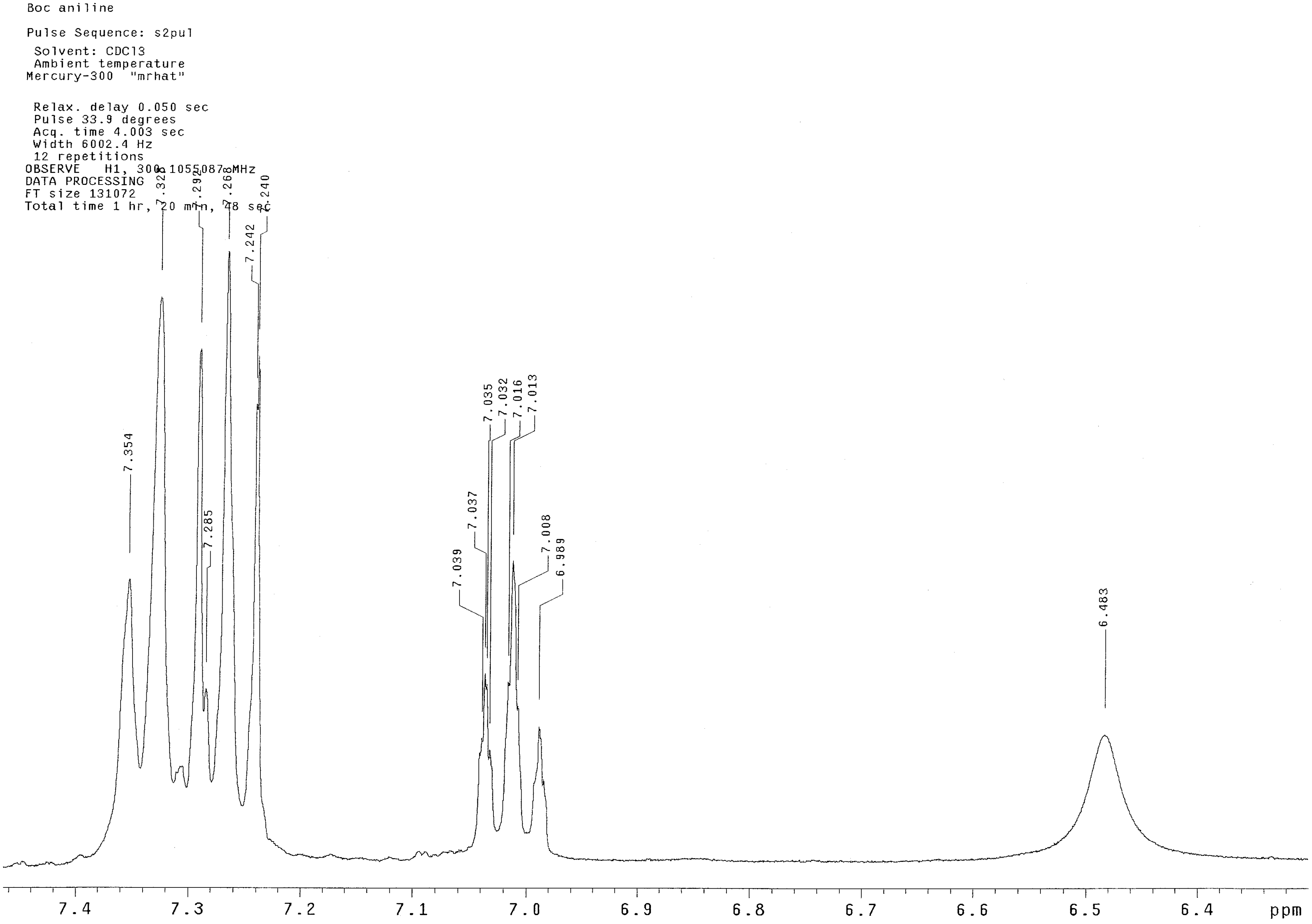
$$\begin{aligned} K_{\max} &= D + \frac{v_{\max} r^2}{48D} = 2{\times}10^{-3}\ \mathrm{mm}^2/\mathrm{s} + \frac{12.5\ \mathrm{mm/s}(0.2\ \mathrm{mm})^2}{48 \cdot 2{\times}10^{-3}\ \mathrm{mm}^2/\mathrm{s}}, \\ &= 5.2103\ \mathrm{mm}^2/s, \\ &= 2.6052{\times}10^3 D, \end{aligned}$$

which is about 30 times smaller than the numerical diffusion estimate. Therefore, we would expect the effective dispersion in the simulation to be about 30 times greater than the dispersion in the true system.

The coefficient of thermal expansion of all solutions was taken to be $0.75{\times}10^{-3}\ \mathrm{K}^{-1}$ as measured in [64] for pure DMF.

4.6.2 ^{1}H-NMR spectra for crude N-Boc-aniline product from batch reaction

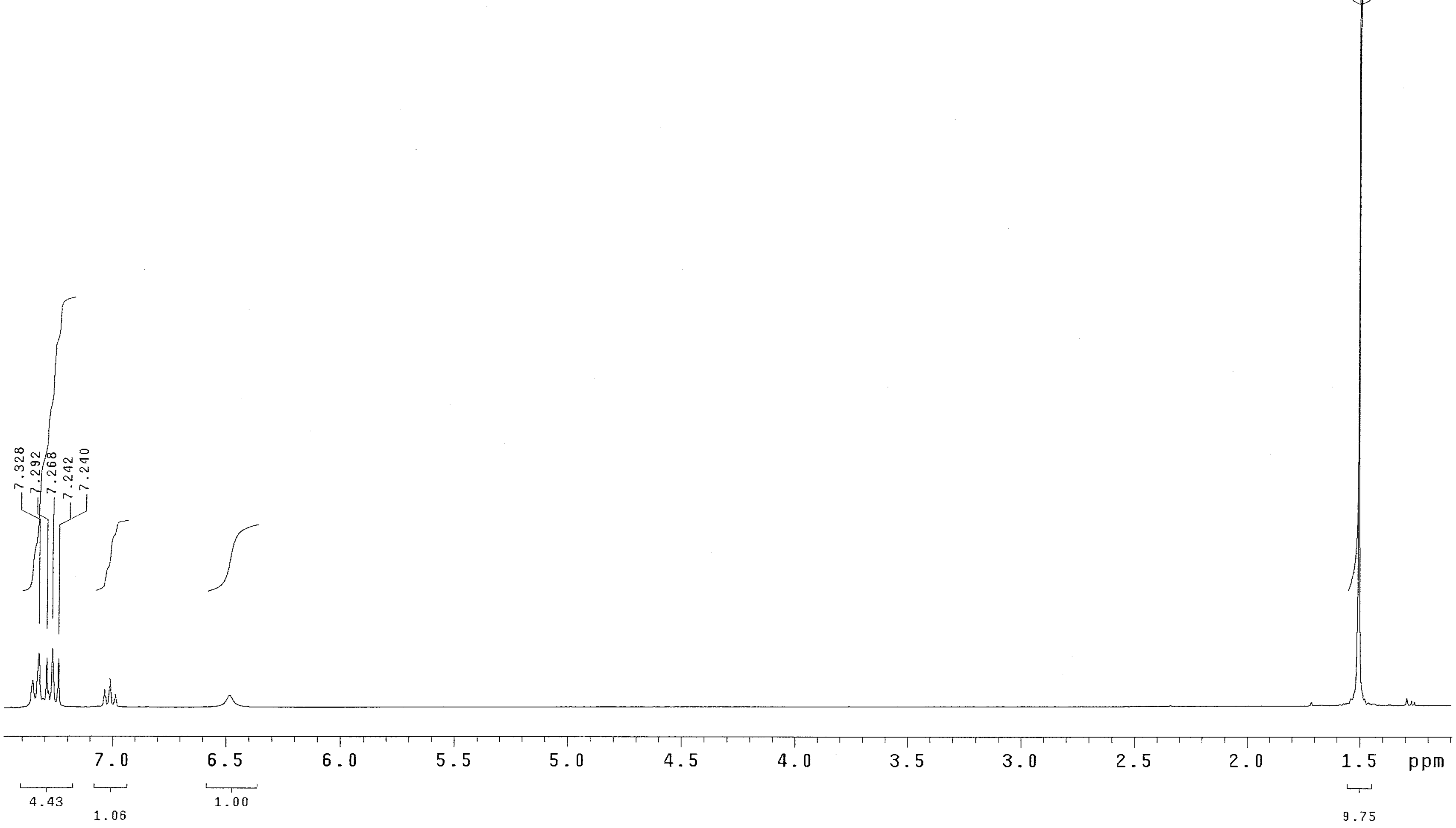




```
Boc aniline

Pulse Sequence: s2pul
 Solvent: CDCl3
 Ambient temperature
Mercury-300  "mrhat"

 Relax. delay 0.050 sec
 Pulse 33.9 degrees
 Acq. time 4.003 sec
 Width 6002.4 Hz
 12 repetitions
OBSERVE   H1, 300.1055087 MHz
DATA PROCESSING
FT size 131072
Total time 1 hr, 20 min, 48 sec
```

Boc aniline

```
Pulse Sequence: s2pul
 Solvent: CDCl3
 Ambient temperature
Mercury-300  "mrhat"

 Relax. delay 0.050 sec
 Pulse 33.9 degrees
 Acq. time 4.003 sec
 Width 6002.4 Hz
 12 repetitions
OBSERVE   H1, 300.1055087 MHz
DATA PROCESSING
FT size 131072
Total time 1 hr, 20 min, 48 sec
```



Chapter 5

Conclusions and outlook

5.1 Summary of contributions

The two most significant contributions in this thesis are the software for deterministic global dynamic optimization (GDO) (Chapter 2) and the convergence-order analysis of auxiliary-ODE-based bounds and relaxations of the solutions of ODEs (Chapter 3). The GDO software, named dGDOpt, gives CPU times up to 90 times faster than methods published in 2006 by Singer and Barton [187] and up to 60 times faster than recently published methods from Sahlodin's thesis [162] on certain problems. In Chapter 3, it was shown for the first time that the bounds [186] and relaxations [174, 177] on the solutions of ODEs have first- and second-order convergence, respectively, under mild assumptions. It was also shown for the first time that certain of these computationally efficient bounds and relaxations can shed conservatism over time, for example, if the bounds or relaxations at the initial condition are overly conservative. In contrast, it was also shown that other, more naïve, methods for computing bounds and relaxations can only become looser as time goes on, as pointed out by [174].

In Chapter 4, optimal experimental design (OED), also known as model-based design of experiments, was applied to design time-varying experiments for microreactor systems for estimation of kinetic model parameters and discrimination between candidate kinetic models. To our knowledge, time-varying experiments for microreactors have never been designed using OED techniques nor have time-varying experiments as general as those here been used to estimate parameters or distinguish between models in microreactors. The dynamic simulations embedded in the parameter estimation problems had up to 1,422 state variables and 57 control parameters and those embedded in the experimental design optimization problems had up to 11,845 state variables (2,776 differential).

5.2 Outlook

Following on from this thesis, there are plans to extend dGDOpt to be able to solve problems with differential algebraic equations (DAEs) embedded, making it applicable to more general dynamic systems. There are also plans to develop and test the performance of new relaxation methods for ODEs. With the series of refinements that have been made in this thesis to the branch-and-bound (or, more aptly, branch-and-reduce) implementation and the results shown when using a linear programming solver for the lower-bounding problem and the relative efficiencies of the different bounding and relaxation methods in the ODE case, future workers are positioned to achieve much better results than they would if using a simplistic branch-and-bound routine with bisection on the decision variable with the largest absolute diameter and no domain reduction.

From the convergence analysis work of Chapter 3, it is now clearer why relaxation-preserving dynamics [174] are so far superior to relaxation-amplifying dynamics [177]. It would be instructive to undertake a similar analysis for global dynamic optimization methods based on Taylor models [114–117] and McCormick-Taylor models [162, 163] to show the convergence order and the dependence of the conservatism of the bounding method on time.

Appendix A

Convergence of convex relaxations from McCormick's product rule when only one variable is partitioned (selective branching)

The following lemma is useful if we want to branch on a subset of the problem variables (whose host set is called Y in the lemma) but still achieve linear convergence. The assumptions are similar to, but not the same as, [65, Conditions W]. The following result is valid if Condition W.6(a) holds but is not valid if only Condition W.6(b) holds.

Lemma A.0.1. *Let $X \subset \mathbb{R}^{n_x}$, $Y \subset \mathbb{R}^{n_y}$, $\mathbf{c} \subset \mathbb{R}^{n_x}$, $v_X : X \ni \mathbf{x} \mapsto \mathbf{c}^{\mathrm{T}}\mathbf{x} \in \mathbb{R}$, $v_Y : Y \to \mathbb{R}$. We use v_X as the scheme of estimators for itself, since it is convex and concave. Let v_Y be locally Lipschitz on Y. Let V_X^B, the inclusion function for v_X, be inclusion monotonic and let V_Y^B, the inclusion function for v_Y, have Hausdorff convergence of order at least 1. Let V_Y^C, the scheme of estimators for v_Y, have pointwise convergence of order at least 1 in Y. Let $g : X \times Y \to \mathbb{R} : (\mathbf{x}, \mathbf{y}) \mapsto v_X(\mathbf{x})v_Y(\mathbf{y})$. Let $G^C \equiv [g^{cv}, g^{cc}]$ be the relaxations of g given by the scheme of McCormick [125]. Then for every compact $Y' \subset Y$ and every $X' \in \mathbb{I}X$,*

$\exists \tau \in \mathbb{R}_+$ *such that* G^C *satisfies*

$$\sup_{\mathbf{x}\in\widehat{X},\mathbf{y}\in\widehat{Y}} w\Big(G^C((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))\Big) \leq \tau w\Big(\widehat{Y}\Big), \quad \forall(\widehat{X},\widehat{Y}) \in \mathbb{I}X' \times \mathbb{I}Y'.$$

Proof. Choose any compact $Y' \subset Y$. Choose any $\widehat{X} \in \mathbb{I}X$, $\widehat{Y} \in \mathbb{I}Y'$ and $(\mathbf{x},\mathbf{y}) \in \widehat{X} \times \widehat{Y}$. We can combine inequalities (4), (5), and (8) from [32, Proof of Theorem 4] to obtain:

$$\begin{aligned}&|g(\mathbf{x},\mathbf{y}) - g^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))|\\ &\leq (v_X(\mathbf{x}) - v_X^L(\widehat{X}))(v_Y(\mathbf{y}) - v_Y^L(\widehat{Y}))\\ &\quad + |v_X^L(\widehat{X})|\max\{v_Y(\mathbf{y}) - v_Y^{cv}((\widehat{Y},[\mathbf{y},\mathbf{y}])), v_Y^{cc}((\widehat{Y},[\mathbf{y},\mathbf{y}])) - v_Y(\mathbf{y})\}\\ &\quad + |v_Y^L(\widehat{Y})|\max\{v_X(\mathbf{x}) - v_X^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}])), v_X^{cc}((\widehat{X},[\mathbf{x},\mathbf{x}])) - v_X(\mathbf{x})\}.\end{aligned} \tag{A.1}$$

Since $v_X(\mathbf{x}) \in V_X^B(\widehat{X})$ and $v_Y(\mathbf{y}) \in V_Y^B(\widehat{Y})$, we have:

$$(v_X(\mathbf{x}) - v_X^L(\widehat{X}))(v_Y(\mathbf{y}) - v_Y^L(\widehat{Y})) \leq w\Big(V_X^B(\widehat{X})\Big)w\Big(V_Y^B(\widehat{Y})\Big). \tag{A.2}$$

Similarly we know that the factors are bounded by the schemes of estimators, $v_X(\mathbf{x}) \in V_X^C((\widehat{X},[\mathbf{x},\mathbf{x}]))$ and $v_Y(\mathbf{y}) \in V_Y^C((\widehat{Y},[\mathbf{y},\mathbf{y}]))$, so that:

$$\begin{aligned}&|v_X^L(\widehat{X})|\max\{v_Y(\mathbf{y}) - v_Y^{cv}((\widehat{Y},[\mathbf{y},\mathbf{y}])), v_Y^{cc}((\widehat{Y},[\mathbf{y},\mathbf{y}])) - v_Y(\mathbf{y})\} \leq |v_X^L(\widehat{X})|w\Big(V_Y^C((\widehat{Y},[\mathbf{y},\mathbf{y}]))\Big),\\ &|v_Y^L(\widehat{Y})|\max\{v_X(\mathbf{x}) - v_X^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}])), v_X^{cc}((\widehat{X},[\mathbf{x},\mathbf{x}])) - v_X(\mathbf{x})\} \leq |v_Y^L(\widehat{Y})|w\Big(V_X^C((\widehat{X},[\mathbf{x},\mathbf{x}]))\Big).\end{aligned} \tag{A.3}$$

By substituting (A.2) and (A.3) into (A.1), it follows that

$$\begin{aligned}&|g(\mathbf{x},\mathbf{y}) - g^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))|\\ &\quad\leq w\Big(V_X^B(\widehat{X})\Big)w\Big(V_Y^B(\widehat{Y})\Big) + |v_X^L(\widehat{X})|w\Big(V_Y^C((\widehat{Y},[\mathbf{y},\mathbf{y}]))\Big) + |v_Y^L(\widehat{Y})|w\Big(V_X^C((\widehat{X},[\mathbf{x},\mathbf{x}]))\Big).\end{aligned} \tag{A.4}$$

Since the scheme of estimators for v_X is exact, we know that $w(V_X^C((\widehat{X},[\mathbf{x},\mathbf{x}]))) = 0$, so

(A.4) reduces to

$$\begin{aligned} &|g(\mathbf{x},\mathbf{y}) - g^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))| \\ &\quad \leq w\Big(V_X^B(\widehat{X})\Big) w\Big(V_Y^B(\widehat{Y})\Big) + |v_X^L(\widehat{X})| w\Big(V_Y^C((\widehat{Y},[\mathbf{y},\mathbf{y}]))\Big). \end{aligned} \tag{A.5}$$

Since V_X^B is inclusion monotonic, $w(V_X^B(\widehat{X})) \leq w(V_X^B(X'))$ and $|v_X^L(\widehat{X})| \leq |v_X^L(X')|$ for any $\widehat{X} \in \mathbb{I}X'$, so we have

$$\begin{aligned} &|g(\mathbf{x},\mathbf{y}) - g^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))| \\ &\quad \leq w\big(V_X^B(X')\big) w\Big(V_Y^B(\widehat{Y})\Big) + |v_X^L(X')| w\Big(V_Y^C((\widehat{Y},[\mathbf{y},\mathbf{y}]))\Big). \end{aligned} \tag{A.6}$$

Since v_Y is Lipschitz on the compact set Y' and its inclusion function V_Y^B has Hausdorff convergence at least 1, we know there exist $L, \tau_1 \in \mathbb{R}_+$ such that for whichever $\widehat{Y} \in \mathbb{I}Y'$ we have chosen,

$$\begin{aligned} w\Big(V_Y^B(\widehat{Y})\Big) &\leq w\Big(\bar{v}_Y(\widehat{Y})\Big) + 2d_H\left(\bar{v}_Y(\widehat{Y}), V_Y^B(\widehat{Y})\right) \\ &\leq (L + 2\tau_1) w\Big(\widehat{Y}\Big), \end{aligned} \tag{A.7}$$

where $\bar{v}_Y(\widehat{Y})$ denotes the exact image of $\widehat{Y}$ under v_Y. Substituting (A.7) into (A.6), we have

$$\begin{aligned} &|g(\mathbf{x},\mathbf{y}) - g^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))| \\ &\quad \leq w\big(V_X^B(X')\big)(L + 2\tau_1) w\Big(\widehat{Y}\Big) + |v_X^L(X')| w\Big(V_Y^C((\widehat{Y},[\mathbf{y},\mathbf{y}]))\Big). \end{aligned} \tag{A.8}$$

Next, use the known pointwise convergence of the scheme for v_Y, which means $\exists \tau_2 \in \mathbb{R}_+$ such that for whichever $\widehat{Y} \in \mathbb{I}Y$ we have chosen,

$$\sup_{\mathbf{y} \in \widehat{Y}} w\Big(V_Y^C((\widehat{Y},[\mathbf{y},\mathbf{y}]))\Big) \leq \tau_2 w\Big(\widehat{Y}\Big), \tag{A.9}$$

so that (A.8) becomes

$$\begin{aligned} &|g(\mathbf{x},\mathbf{y}) - g^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))| \\ &\quad \leq w\big(V_X^B(X')\big)(L+2\tau_1)w\Big(\widehat{Y}\Big) + |v_X^L(X')|\tau_2 w\Big(\widehat{Y}\Big). \end{aligned} \tag{A.10}$$

Take $\tau_3 = w(V_X^B(X'))(L+2\tau_1) + |v_X^L(X')|\tau_2$, so that (A.10) becomes

$$|g(\mathbf{x},\mathbf{y}) - g^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))| \leq \tau_3 w\Big(\widehat{Y}\Big). \tag{A.11}$$

Since the constant τ_3 does not depend on $\mathbf{x}$ or $\mathbf{y}$, we can take the supremum

$$\sup_{\mathbf{x}\in\widehat{X},\mathbf{y}\in\widehat{Y}} |g(\mathbf{x},\mathbf{y}) - g^{cv}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}]))| \leq \tau_3 w\Big(\widehat{Y}\Big). \tag{A.12}$$

Note also that τ_3 does not depend on $\widehat{X}$ or $\widehat{Y}$, and a similar result holds for

$$\sup_{\mathbf{x}\in\widehat{X},\mathbf{y}\in\widehat{Y}} |g^{cc}((\widehat{X},[\mathbf{x},\mathbf{x}]),(\widehat{Y},[\mathbf{y},\mathbf{y}])) - g(\mathbf{x},\mathbf{y})|,$$

so the result follows. □

Linear convergence in the case of selective branching is the strongest result that can be proven, as the following example shows.

Example A.0.2. Consider the McCormick relaxation of $f : X \times Y : (x,y) \mapsto xy$:

$$\begin{aligned} &F^C(([x^L,x^U],[y^L,y^U]),([x,x],[y,y])) \\ &= [\max\{y^L x + x^L y - x^L y^L, y^U x + x^U y - x^U y^U\}, \\ &\quad \min\{y^U x + x^L y - x^L y^U, y^L x + x^U y - x^U y^L\}]. \end{aligned}$$

Figure A-1 shows the linear pointwise convergence of F^C when only the Y space is partitioned, using $\widehat{X} = [-2.5, 12.5]$ and $\widehat{Y} = [-2.5-\varepsilon, -2.5+\varepsilon]$. It can be seen that $\sup_{(x,y)\in\widehat{X}\times\widehat{Y}} w(F^C((\widehat{X},[x,x]),(\widehat{Y},[y,y]))) = 7.5\varepsilon = 0.5w(\widehat{X})\varepsilon = w(\widehat{X})w(\widehat{Y})$. See also the proof of Lemma 3.9.19, where the bound $w(\widehat{X})w(\widehat{Y})$ is also shown, but used for a different final result.

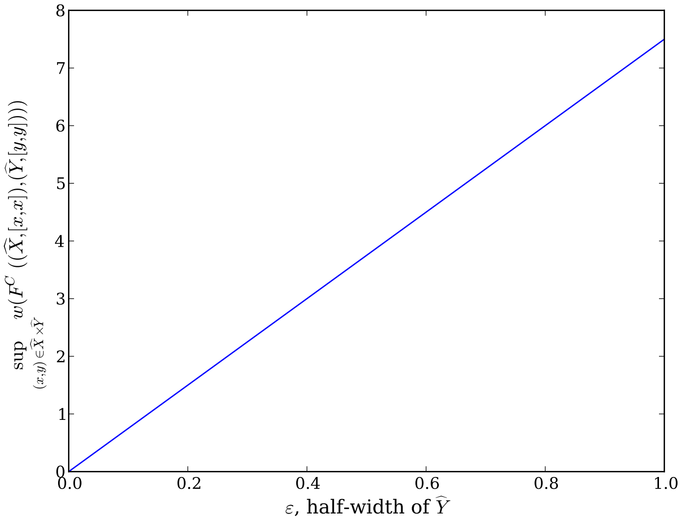


Figure A-1: Example A.0.2 shows linear pointwise convergence of the bilinear form when only one variable is partitioned.

Appendix B

Economic analysis of integrated continuous and batch pharmaceutical manufacturing: a case study

This chapter was joint work with Dimitrios I. Gerogiorgis, Rohit Ramachandran, James M. B. Evans, Paul I. Barton, and Bernhardt L. Trout, and was published in [165].

Abstract

The capital, operating, and overall costs of a dedicated continuous manufacturing process to synthesize an active pharmaceutical ingredient (API) and formulate it into tablets are estimated for a production scale of 2000 metric tons of tablets per year, with raw material cost, production yield, and API loading varied over broad ranges. Costs are compared to batch production in a dedicated facility. Synthesis begins with a key organic intermediate three synthetic steps before the final API; results are given for key intermediate (KI) costs of \$100 to \$3000/kg, with drug loadings in the tablet of 10 and 50 wt%. The novel continuous process described here is being developed by an interdisciplinary team of 20 researchers.

Since yields are not yet well-known, and continuous processes typically have better yields than batch ones, the overall yields of the continuous processes with recycling were set equal to that of the batch process. Without recycling, yields are 10% lower, but less equipment is required. The continuous process has not been built at large scale, so Wroth factors and other assumptions were used to estimate costs. Capital expenditures for continuous production were estimated to be 20 to 76% lower, depending on the drug loading, KI cost, and process chosen; operating expenditures were estimated to be between 40% lower and 9% higher. The novel continuous process with recycling coupled to a novel direct tablet formation process yields the best overall cost savings in each drug loading/KI price scenario: estimated savings range from 9 to 40%. Overall cost savings are also given assuming the yield in the continuous case is 10% above and 10% below that of the batch process. Even when yields in the continuous case are lower than in the batch case, savings can still be achieved because the labor, materials handling, CapEx, and other savings compensate.

B.1 Introduction

Continuous manufacturing (CM) is attracting increasing attention within the pharmaceutical industry today because it could lead to significant decreases in production costs while improving product quality [126, 151]. Historically, production costs were seen as a small enough part of the overall industry expenses that major cost reductions were not needed. Regulations also drove production towards the batch mode, since processes were required to be run in exactly the same way for the lifetime of the therapy. Also, batch production allows verification of quality of each batch from each process before further processing, whereas a “batch” in a continuous process is not contained in the same way [112, 208]. Today, however, it is becoming more difficult for pharmaceutical companies to meet profit expectations, due to increasing research and development (R&D) costs and competition from generics manufacturers [17]. At the same time, regulatory bodies are shifting the emphasis towards process understanding and giving more freedom when such understanding is demonstrated [89]. For sufficiently large production scales, continuous processes tend to have lower production costs; CM would also allow manufacturers to use the increased process understanding for on-line process control, yielding consistently high-quality product and less material wasted as off-spec product [99, 155].

A review of the fine and commodity chemical industries demonstrates that CM could offer both operating expenditure (OpEx) and capital expenditure (CapEx) savings for the pharmaceutical industry. Labor for transporting material between batch units, labor for quality assurance/quality control (QA/QC), and in-process inventory (working capital) can all be significantly reduced in continuous processing [151, 154]. Processing equipment for fine chemical synthesis can be made much smaller by moving to continuous processing, as well has having larger surface area to volume ratios, which implies a safer plant (a smaller holdup of solvents in reactors and enhanced heat transfer for safe handling of highly exothermic reactions), a smaller investment in reactors, and faster change over in multipurpose plants [99, 107, 155]. More rapid mixing, reaction, and quenching are possible in continuous flow [100], enabling reactions that would produce significantly more impurities if run in batch mode, such as in the first reaction in the novel continuous process presented in this work. Plant footprint can also be reduced due to smaller processing equipment, with commensurate energy savings for heating, ventilation, and air conditioning [199].

Pharmaceutical processes often contain continuous or semi-continuous processing steps, such as milling and tablet compression, but the processes are started and stopped to mirror the batch processing in other steps. These steps can be more naturally run in a continuous manner, potentially yielding more consistent product quality [151, 208]. Scaleup of batch granulation can be difficult, and is sometimes easier in continuous mode, so development of a needed granulation process could begin on continuous equipment, easing scaleup for production [112]. Recently, the lack of continuous tablet coating equipment was a bottleneck for continuous pharmaceutical production [208], but now it is available. Continuous powder mixing has been shown to perform as needed, with excellent time stability [23].

In addition to cost savings, developing continuous processes early on, using microreactors for instance, can enhance process understanding early in the patent life of a product, easing scale-up and leading to additional time during which the product can be sold exclusively by the patent holder, as well as the ability to bring therapies to ailing people more quickly [151]. Recent developments in process analytical technology (PAT) will allow manufacturers to complete the shift to continuous manufacturing, as long as it proves cost-effective [17].

Despite studies on the individual differences between batch and continuous processing [151, 154, 155, 199], to date, an integrated analysis of the continuous manufacture of a final drug product from a late-stage organic KI has not been published. The Novartis-MIT Center for Continuous Manufacturing (CCM) is focused on a holistic approach where we consider manufacture of the final drug product from starting materials available as fine chemicals. In this work we estimate CapEx, OpEx, and present cost of a dedicated batch process and four continuous processes that are enabled by new technologies developed for continuous production. While many pharmaceutical production processes use multi-purpose equipment to manufacture several drugs in partial-year campaigns, very high-volume drugs are sometimes produced on dedicated equipment.

B.2 Process description

For both the batch and continuous processes, the assessment starts with a late-stage organic key intermediate (KI) molecule, three synthetic steps before the final active pharmaceutical ingredient (API), and produces a final drug product: tablets. The production scale is 2000 metric tons of tablets per year, which is on par with the production scale of a very high-volume "blockbuster" drug. API loadings in the tablet of 10 and 50 wt% were used to account for variations in API potency. Both processes produce the same drug product. The batch process has been extensively developed by Novartis, whereas the continuous process is being developed in the Center for Continuous Manufacturing.

B.2.1 Batch process

The sequence of unit operations for the batch process is given in Figure B-1. The raw materials requirements and costs for one scenario are given in Table B.1 and Table B.2. We are not permitted to disclose further details of the process.

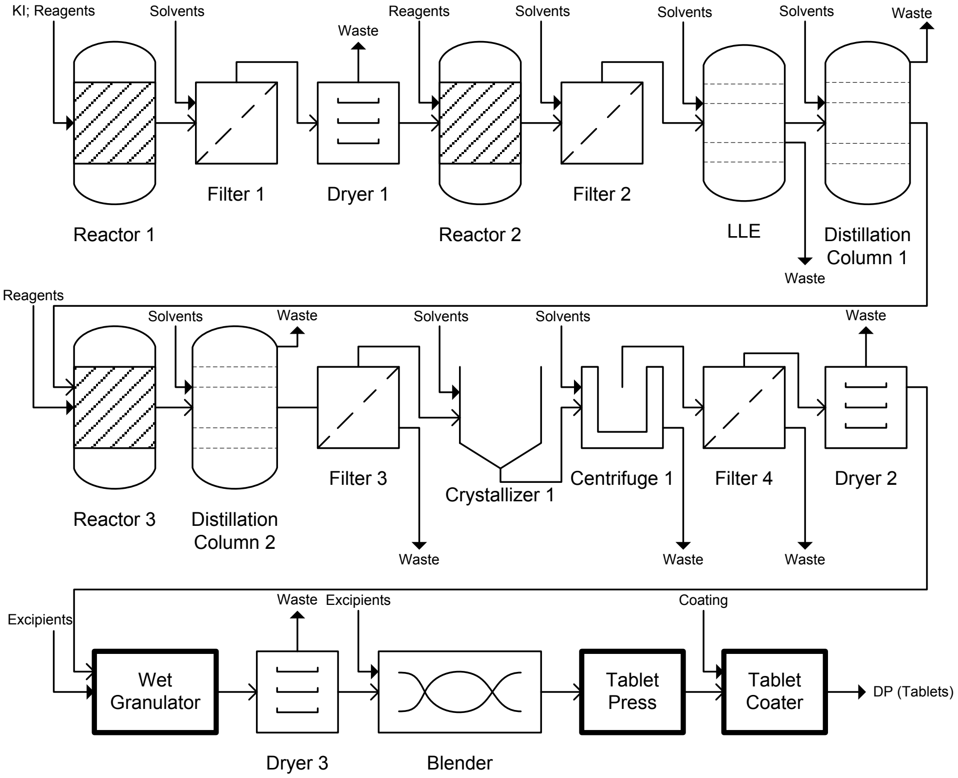


Figure B-1: Process flow diagram for batch (Bx) manufacturing route

B.2.2 Novel CM route (CM1)

The CCM team developed a new synthetic route (CM1; Figure B-2) that utilizes pathways that are not feasible in a batch process. For example, processing in a continuous-flow reactor enables a much more rapid deprotection reaction than batch reaction; translating the CM1 route into a standard batch process would result in significant degradation of the product because the required rate of reagent addition cannot be achieved in batch mode. Also, the continuous processes save an average of 61% of the annual water usage and 21 wt% of the annual solvent usage compared to batch. Reactors 1, 2, and 3 are plug-flow reactors. The crystallizer and combined reactor/crystallizer are agitated tanks. The API synthesis is coupled with two downstream process options: roller compaction (RC) and a novel direct tablet formation (DTF) process. RC is a well-established pharmaceutical technology; a patent application is being prepared for the novel direct tablet formation process, so it is not described in detail here. Since the yields for the final continuous processes are not known precisely, yields have been set such that the overall yield for the continuous process with recycling for the first reaction (CM1R) is equal to that of the batch process, and the overall yield of the continuous process without recycling (CM1) is 10% below that of the batch process. Overall cost savings are also given for the case where overall yield for process CM1R is 10% below and 10% above that of the batch case. In each case, the overall yield for process CM1 is 10% below the corresponding yield for process CM1R. The actual yields that have been demonstrated in bench-scale continuous reactions are bracketed by these scenarios; it is believed that a mature continuous process will have yields equal to or better than the batch yields, since the continuous process already has competitive yields despite being developed for fewer than half as many years, at a much smaller scale, and by fewer people.

Continuous reactions scale up very predictably and in an economically favorable way [155]. One issue currently limiting the savings is microreactor plugging or fouling, which can be observed as an increasing pressure drop across the reactor [99, 155]. The methods for using microreactors with heterogeneous catalysts or severe precipitation are not mature [99]. However, several workarounds to the plugging and fouling issue are possible, based on including strategic solvent choice, flow velocity, temperature, and device geometry [99]. Microreactors have been successfully used to produce hundreds of kg of product in a few weeks [155].

B.2.3 Novel CM route with recycle (CM1R)

Process CM1R is identical to CM1, except that a single recycle loop and appropriate separation equipment are added to increase the effective yield (from 86.4% to 98.5%) in the first step (Reactor 1) of the API synthesis from the KI. Separation equipment and recycle are essential in order to reduce formation of the primary impurity. Without the separation step, the primary product of the reaction can undergo a subsequent reaction to form an impurity. The overall yields (mol drug substance/mol KI) of processes CM1 and CM1R are 69% and 79%, respectively; that of the batch process is 79%. Even without recycling, significant savings overall are estimated, due to savings in CapEx, working capital, quality assurance and control, labor, materials handling, waste handling, and utilities.

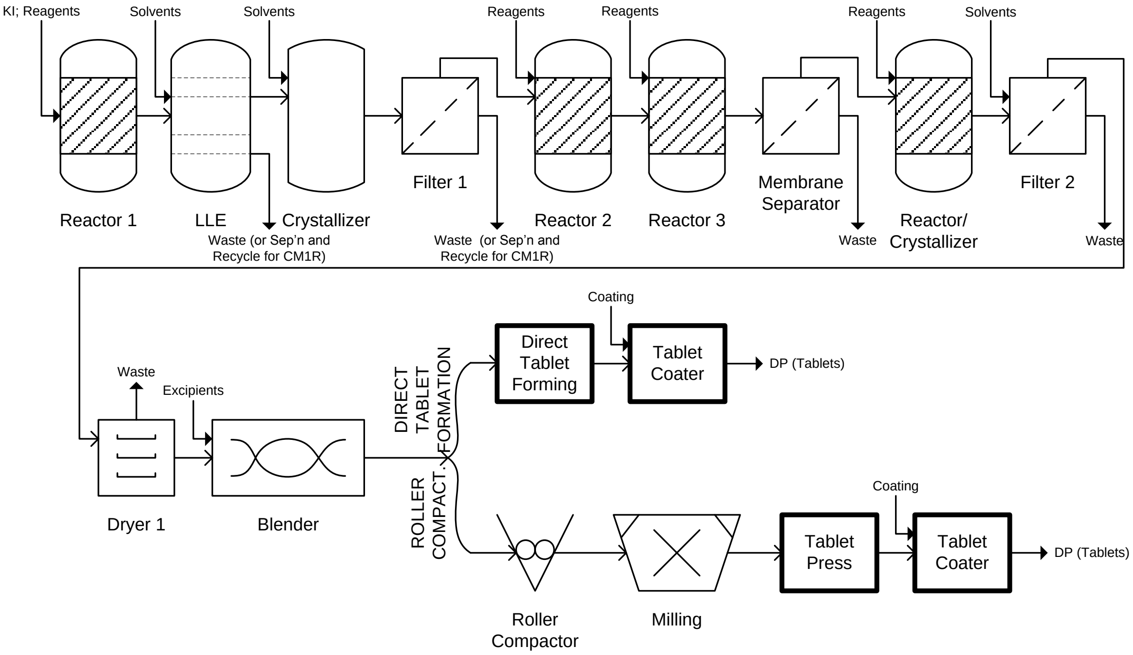


Figure B-2: Process flow diagram for continuous manufacturing route CM1, showing both options for forming tablets

Table B.1: Raw materials requirements for all processes at 50 wt% API loading

Materials	Bx	CM1R/DTF	CM1/DTF	CM1R/RC	CM1/RC
Organic reagents	1,955,000	1,597,000	3,112,000	1,597,000	3,112,000
Inorganic reagents	5,508,000	3,659,000	3,659,000	3,659,000	3,659,000
Organic solvents	34,090,000	24,659,000	29,497,000	24,659,000	29,497,000
Water	22,907,000	7,803,000	9,965,000	7,803,000	9,965,000
Excipients and coatings	1,004,000	1,000,000	1,000,000	1,000,000	1,000,000
Total	65,464,000	38,718,000	47,233,000	38,718,000	47,233,000

All values in kg/yr. DTF: direct tablet formation; RC: roller compaction.

B.2.4 Material balances

Material requirements and costs for all processes are given in Tables B.1 and B.2. For Table B.2, the cost of the KI (one of the organic reagents) is $3000 kg/yr, whereas costs for other raw materials are from vendor quotes, and are typically much less than $3000/kg.

B.3 Cost analysis methods

Green-field construction of a new, dedicated plant was considered in all cases. A 335-day working year was considered, with 30 days left for maintenance, cleaning, and startup/shutdown.

Table B.2: Raw materials costs for all processes at 50 wt% API loading and \$3000/kg KI

Materials	Bx	CM1R/DTF	CM1/DTF	CM1R/RC	CM1/RC
Organic reagents	3,394,145,000	3,375,898,000	3,899,888,000	3,375,898,000	3,899,888,000
Inorganic reagents	2,674,000	4,784,000	4,784,000	4,784,000	4,784,000
Organic solvents	92,356,000	22,864,000	27,263,000	22,864,000	27,263,000
Water	2,182,000	780,000	996,000	780,000	996,000
Excipients and coatings	15,936,000	15,893,000	15,893,000	15,893,000	15,893,000
Total	3,507,293,000	3,420,219,000	3,948,824,000	3,420,219,000	3,948,824,000

All values in \$/yr. DTF: direct tablet formation; RC: roller compaction

One production line per plant was assumed. Batch process effective utilization time was assumed to be 85% for upstream processes and 55% for downstream processes; 95% was assumed for all continuous processes. This is the percentage of time when the process equipment is actually processing material. The remaining time is spent filling, emptying, and cleaning the batch processing unit, or simply waiting for material to be processed. These assumptions are optimistic for batch production, representing lean batch operations in dedicated production: According to Vervaet and Remon [208] the overall equipment effectiveness (OEE), a related metric, takes a typical value in batch pharmaceutical production of 30%, with good processes having 74% and “best-in-class” production lines reaching 92%.

B.3.1 Capital expenditures (CapEx)

B.3.1.1 Equipment size and cost estimation

Vendor price quotations for all process equipment were obtained for both batch and continuous equipment over a wide range of sizes, and the smallest unit of sufficiently large size was selected. When price quotations were only available for batch equipment, a 10% price premium was assumed for continuous units relative to a batch unit of the same size, to account for the increased process engineering (CapEx) required to operate a process continuously with feedback control, as compared to batch processes which are typically operated in an open-loop manner. Scaling of cost could be approximated well ($R^2 \geq 0.98$) by a power law in the following cases: plug-flow reactor, exponent 0.42; filtration equipment, exponent 0.33; agitated vessel/CSTR/crystallizer, exponent 0.20; dryer, exponent 0.21.

Table B.3: Selected Wroth Factors [54]

Unit	Wroth factor
Distillation tower and internals	4.0
Instrument	4.1
Process tank	4.1
Reactor	(factor into appropriate process tanks and other equipment)
Storage tank	3.5
All other equipment	3.5

B.3.1.2 Calculation of overall CapEx from individual process equipment costs

The total cost of processing equipment excluding any ancillary equipment, delivery, electrical, engineering, or piping expenses is termed the FOB (free on board) cost. From delivered equipment cost (1.05x FOB cost), Wroth factors (Table B.3) were used to calculate delivered installed equipment cost, which includes ancillary equipment, delivery, electrical, engineering, and piping costs [54]. Wroth factors allow quick estimation of installation and other necessary equipment costs, and are commonly used at this stage of an economic analysis.

$$\text{(Delivered installed equipment cost)} = \text{(Wroth factor)} \times \text{(Delivered equipment cost)} \quad \text{(B.1)}$$

Additional CapEx heuristics used in the present analysis are summarized in Table B.4. Since pharmaceutical production scales are smaller than typical commodity chemical production scales, and must adhere to stricter hygiene regulations, the additional expenses are expected to comprise a larger fraction of the CapEx. Thus, the values used were the upper bounds of the ranges given by [54].

B.3.2 Operating expenditures (OpEx)

Operating expenditures were calculated for KI prices of \$100, \$500, and \$3000/kg. The heuristics used are summarized in Table B.5. The continuous plant has not been built at large scale, so the values for labor, materials handling, and QA/QC savings represent our best estimates at this time. For example, OpEx savings are expected in QA/QC since some manual sampling and analysis can be replaced by on-line analysis.

Table B.4: Summary of CapEx Heuristics Used

Item	Cost
(1) FOB cost	Sum of processing equipment units [54]
(2) Delivery	5% of FOB cost [54]
(3) Installation: ancillary equipment, automation, electrical, piping, and engineering	[(Wroth factor)−1]x(delivered equipment cost) [54]
(4) Battery-limits installed cost (BLIC)	Sum of items (1) to (3) [54]
(5) Buildings and structures	20% of BLIC [54]
(6) Contingency	20% of BLIC [54]
(7) Offsite capital (for a grass-roots plant)	150% of BLIC [54]
(8) Service facilities	20% of BLIC [54]
(9) Waste disposal	Not included in CapEx; assumed to be treated at a nominal cost indicated in Table 5
(10) Working capital	35% of annual sales [54] $\Longrightarrow$ used 35% of annual materials costs for batch; 3.5% for continuous, since throughput times are expected to be 10x lower in continuous processing
(11) Total CapEx	Sum of items (4) to (10)

Table B.5: Summary of OpEx Heuristics Used

Item	Cost
(1) Labor and supervision	$160,000/yr per operator [150]; number of operators estimated as in [204]; twice as many operators required for batch processes as for continuous
(2) Materials handling and storage	Continuous is estimated at 40% of batch
(3) Off-spec product	0% for batch and continuous
(4) Quality assurance and control (QA/QC)	Continuous is estimated at 50% of batch
(5) Utilities	$1.50/kg material input
(6) Waste disposal	$2.50/gallon for water and organic solvents; $15.00/gallon for all other material [53]
(7) Total OpEx	Sum of items (1) to (6) plus raw material costs

B.3.3 Overall cost of production

To quantify overall cost differences accounting for both CapEx and OpEx, present cost of the project (B.2) was calculated for each processing option. This is the discounted total cost of the project, excluding any revenue. Present cost of the project is similar to net present value (NPV) (B.3), but does not include revenue. This figure of merit was chosen because we are comparing costs, not NPV.

$$\text{(Present Cost)} = \text{(CapEx)} + \sum_{i=1}^{\tau} \frac{\text{(OpEx)}}{(1+r_d)^i} \tag{B.2}$$

$$\text{(NPV)} = -\text{(CapEx)} + \sum_{i=1}^{\tau} \left\{ \frac{-\text{(OpEx)}}{(1+r_d)^i} + \frac{\text{(Revenue)}}{(1+r_d)^i} \right\} \tag{B.3}$$

Discount rate (r_d) was 7%, construction period was 1 year, and project lifetime (τ) was 15 years.

B.3.4 Contributors to overall cost savings

To quantify the contributions of different expenses to the cost differences for CM relative to batch, the following quantity was defined:

$$\begin{pmatrix} \text{Contribution to present} \\ \text{cost difference} \end{pmatrix} = \frac{\begin{pmatrix} \text{Present cost of contributor} \\ \text{for Bx process} \end{pmatrix} - \begin{pmatrix} \text{Present cost of contributor} \\ \text{for CM process} \end{pmatrix}}{\text{(Present cost of Bx process)}}. \tag{B.4}$$

To clarify the above definition, note that

$$\sum_{\text{(contributors)}} \begin{pmatrix} \text{Contribution to present} \\ \text{cost difference} \end{pmatrix} = \begin{pmatrix} \text{Percentage present cost} \\ \text{difference vs. batch} \end{pmatrix}. \tag{B.5}$$

B.4 Results

B.4.1 Capital expenditures (CapEx)

Process CM1 with direct tablet formation has the largest CapEx savings (31–76% savings vs. batch processing). At the highest KI price, the working capital, especially for the KI,

Table B.6: CapEx (including working capital) differences for all process options, relative to batch case, for upstream and downstream

Cost of KI	$100/kg	$500/kg	$3000/kg	$100/kg	$500/kg	$3000/kg
API loading	10 wt%	10 wt%	10 wt%	50 wt%	50 wt%	50 wt%
Batch	Tot: [$315M]	Tot: [$346M]	Tot: [$542M]	Tot: [$429M]	Tot: [$585M]	Tot: [$1565M]
	U: [$73M]	U: [$105M]	U: [$300M]	U: [$173M]	U: [$329M]	U: [$1308M]
	D: [$242M]	D: [$242M]	D: [$242M]	D: [$256M]	D: [$256M]	D: [$256M]
CM1R,	Tot: -28%	Tot: -33%	Tot: -54%	Tot: -39%	Tot: -53%	Tot: -76%
DTF	U: -31%	U: -49%	U: -76%	U: -52%	U: -70%	U: -85%
	D: -27%	D: -27%	D: -27%	D: -31%	D: -31%	D: -31%
CM1,	Tot: -31%	Tot: -36%	Tot: -55%	Tot: -42%	Tot: -55%	Tot: -76%
DTF	U: -43%	U: -57%	U: -78%	U: -59%	U: -73%	U: -85%
	D: -27%	D: -27%	D: -27%	D: -31%	D: -31%	D: -31%
CM1R,	Tot: -20%	Tot: -26%	Tot: -49%	Tot: -34%	Tot: -49%	Tot: -75%
RC	U: -31%	U: -49%	U: -76%	U: -52%	U: -70%	U: -85%
	D: -17%	D: -17%	D: -17%	D: -21%	D: -21%	D: -21%
CM1,	Tot: -23%	Tot: -29%	Tot: -50%	Tot: -36%	Tot: -50%	Tot: -74%
RC	U: -43%	U: -57%	U: -78%	U: -59%	U: -73%	U: -85%
	D: -17%	D: -17%	D: -17%	D: -21%	D: -21%	D: -21%

All percentage differences are relative to Batch (top row). DTF: direct tablet formation; RC: roller compaction; Tot: total CapEx; U: upstream CapEx; D: downstream CapEx. CapEx dollar amounts are provided in square brackets for the base case of a batch process.

Table B.7: Summary of CapEx differences for all process options, relative to batch case

Cost of KI	$100/kg	$500/kg	$3000/kg	$100/kg	$500/kg	$3000/kg
API loading	10 wt%	10 wt%	10 wt%	50 wt%	50 wt%	50 wt%
Batch (basis for differences)	[$315M]	[$346M]	[$542M]	[$429M]	[$585M]	[$1565M]
CM1R with direct tablet formation	-28%	-33%	-54%	-39%	-53%	-76%
CM1 with direct tablet formation	-31%	-36%	-55%	-42%	-55%	-76%
CM1R with roller compaction	-20%	-26%	-49%	-34%	-49%	-75%
CM1 with roller compaction	-23%	-29%	-50%	-36%	-50%	-74%

Total CapEx dollar amounts are provided in square brackets for the base case of a batch process.

dominates CapEx, so savings are similar for all processes. Detailed results are given in Table B.6; summarized results are given in Table B.7.

B.4.2 Operating expenditures (OpEx)

Process CM1R with either direct tablet formation or roller compaction has the lowest annual OpEx of any process option (6–40% savings); process CM1 options show slightly less savings, due to the lower overall yield without recycling. Detailed results are given in Table B.8; summarized results are given in Table B.9.

B.4.3 Overall cost of production

Process CM1R with direct tablet formation has the lowest present cost (9–40% savings). CM1R with roller compaction is the next best option, with very similar savings. See Table B.10.

Table B.8: Annual OpEx differences for all process options, relative to batch case

Cost of KI API loading	$100/kg 10 wt%	$500/kg 10 wt%	$3000/kg 10 wt%	$100/kg 50 wt%	$500/kg 50 wt%	$3000/kg 50 wt%
Batch	Tot: [$136M] URM: [$49M] DRM: [$25M] Oth: [$62M]	Tot: [$226M] URM: [$139M] DRM: [$25M] Oth: [$62M]	Tot: [$785M] URM: [$698M] DRM: [$25M] Oth: [$62M]	Tot: [$531M] URM: [$246M] DRM: [$16M] Oth: [$269M]	Tot: [$979M] URM: [$693M] DRM: [$16M] Oth: [$269M]	Tot: [$3777M] URM: [$3491M] DRM: [$16M] Oth: [$269M]
CM1R, DTF	Tot: -33% URM: -36% DRM: 0% Oth: -45%	Tot: -20% URM: -13% DRM: 0% Oth: -45%	Tot: -6% URM: -2% DRM: 0% Oth: -45%	Tot: -40% URM: -36% DRM: -1% Oth: -47%	Tot: -22% URM: -13% DRM: -1% Oth: -47%	Tot: -6% URM: -2% DRM: -1% Oth: -47%
CM1, DTF	Tot: -19% URM: -6% DRM: 0% Oth: -36%	Tot: -6% URM: 7% DRM: 0% Oth: -36%	Tot: 8% URM: 13% DRM: 0% Oth: -36%	Tot: -22% URM: -6% DRM: -1% Oth: -38%	Tot: -6% URM: 7% DRM: -1% Oth: -38%	Tot: 9% URM: 13% DRM: -1% Oth: -38%
CM1R, RC	Tot: -33% URM: -36% DRM: 0% Oth: -45%	Tot: -20% URM: -13% DRM: 0% Oth: -45%	Tot: -6% URM: -2% DRM: 0% Oth: -45%	Tot: -40% URM: -36% DRM: -1% Oth: -47%	Tot: -22% URM: -13% DRM: -1% Oth: -47%	Tot: -6% URM: -2% DRM: -1% Oth: -47%
CM1, RC	Tot: -19% URM: -6% DRM: 0% Oth: -36%	Tot: -6% URM: 7% DRM: 0% Oth: -36%	Tot: 8% URM: 13% DRM: 0% Oth: -36%	Tot: -22% URM: -6% DRM: -1% Oth: -38%	Tot: -6% URM: 7% DRM: -1% Oth: -38%	Tot: 9% URM: 13% DRM: -1% Oth: -38%

All percentage differences are relative to Batch (top row). DTF: direct tablet formation; RC: roller compaction; Tot: total OpEx; URM: upstream raw materials OpEx; DRM: downstream raw materials OpEx; Oth: all other OpEx. OpEx dollar amounts are provided in square brackets for the base case of a batch process.

Table B.9: Summary of annual OpEx differences for all process options, relative to batch case

Cost of KI API loading	$100/kg 10 wt%	$500/kg 10 wt%	$3000/kg 10 wt%	$100/kg 50 wt%	$500/kg 50 wt%	$3000/kg 50 wt%
Batch (basis for differences)	[$136M]	[$226M]	[$785M]	[$531M]	[$979M]	[$3777M]
CM1R with direct tablet formation	-33%	-20%	-6%	-40%	-22%	-6%
CM1 with direct tablet formation	-19%	-6%	8%	-22%	-6%	9%
CM1R with roller compaction	-33%	-20%	-6%	-40%	-22%	-6%
CM1 with roller compaction	-19%	-6%	8%	-22%	-6%	9%

Annual OpEx dollar amounts are provided in square brackets for the base case of a batch process.

Table B.10: Summary of present cost differences for all process options, relative to batch case

Cost of KI API loading	$100/kg 10 wt%	$500/kg 10 wt%	$3000/kg 10 wt%	$100/kg 50 wt%	$500/kg 50 wt%	$3000/kg 50 wt%
Batch (basis for differences)	[$1515M]	[$2337M]	[$7472M]	[$5117M]	[$9225M]	[$34902M]
CM1R with direct tablet formation	-32%	-22%	-9%	-40%	-24%	-9%
CM1 with direct tablet formation	-21%	-10%	4%	-24%	-9%	5%
CM1R with roller compaction	-30%	-21%	-9%	-40%	-23%	-9%
CM1 with roller compaction	-20%	-9%	4%	-23%	-8%	5%

Present cost is the total discounted cost of the project, excluding any revenue, for the 15-year project lifetime. Present cost dollar amount is provided in square brackets for the base case of a batch process.

Table B.11: Summary of present cost differences if CM1R yield is 10% below batch yield

Cost of KI API loading	\$100/kg 10 wt%	\$500/kg 10 wt%	\$3000/kg 10 wt%	\$100/kg 50 wt%	\$500/kg 50 wt%	\$3000/kg 50 wt%
Batch (basis for differences)	[\$1515M]	[\$2337M]	[\$7472M]	[\$5117M]	[\$9225M]	[\$34902M]
CM1R with direct tablet formation	-28%	-15%	3%	-35%	-14%	4%
CM1 with direct tablet formation	-16%	-1%	18%	-16%	3%	20%
CM1R with roller compaction	-27%	-13%	3%	-34%	-14%	4%
CM1 with roller compaction	-14%	0%	18%	-15%	3%	20%

Present cost is the total discounted cost of the project, excluding any revenue, for the 15-year project lifetime. Present cost dollar amount is provided in square brackets for the base case of a batch process.

Table B.12: Summary of present cost differences if CM1R yield is 10% above batch yield

Cost of KI API loading	\$100/kg 10 wt%	\$500/kg 10 wt%	\$3000/kg 10 wt%	\$100/kg 50 wt%	\$500/kg 50 wt%	\$3000/kg 50 wt%
Batch (basis for differences)	[\$1515M]	[\$2337M]	[\$7472M]	[\$5117M]	[\$9225M]	[\$34902M]
CM1R with direct tablet formation	-35%	-28%	-19%	-44%	-31%	-19%
CM1 with direct tablet formation	-25%	-17%	-7%	-30%	-18%	-6%
CM1R with roller compaction	-33%	-27%	-18%	-44%	-31%	-19%
CM1 with roller compaction	-24%	-16%	-7%	-29%	-17%	-6%

Present cost is the total discounted cost of the project, excluding any revenue, for the 15-year project lifetime. Present cost dollar amount is provided in square brackets for the base case of a batch process.

If the overall yield of process CM1R is 10% below that of the batch process (Table B.11), the overall costs of continuous processing are between 4% higher and 35% lower than batch processing if the best process is chosen for each API loading/KI price scenario. At \$3000/kg for the KI, all continuous processes are estimated to be more expensive than the batch process. If the overall yield is 10% higher for CM1R than for the batch process (Table B.12), 19–35% savings can be achieved vs. batch in all low API loading cases by choosing process CM1R with direct tablet formation.

B.4.4 Contributors to overall cost savings

In Table B.13, the present cost of the project for the baseline case in which CM1R yield is equal to batch yield is broken down into contributions for each category (cf. (B.4)). The values are for the novel continuous process with recycling with direct tablet formation (CM1R/DTF), and are similar to those for the other novel CM options (not published).

The expenditures for the KI and for excipients are the same for CM1R/DTF and batch, since equally-priced excipients are used and the overall yield of drug substance from KI is assumed equal in CM1R and batch. The cost of the other organic reagents needed in the novel continuous process are lower, reducing OpEx in the Other raw materials category. The novel continuous process also has lower solvent usage, reducing costs of other raw materials

Table B.13: Contributions to present cost difference relative to batch for novel continuous process with recycling (CM1R) with direct tablet formation

Cost of KI API loading	\$100/kg 10 wt%	\$500/kg 10 wt%	\$3000/kg 10 wt%	\$100/kg 50 wt%	\$500/kg 50 wt%	\$3000/kg 50 wt%
Organic reagents	-2.2%	-1.4%	-0.4%	-3.2%	-1.8%	-0.5%
KI	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Other raw materials	-8.0%	-5.2%	-1.6%	-11.9%	-6.6%	-1.7%
Excipients	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Labor and materials handling	-4.1%	-2.6%	-0.8%	-5.3%	-3.0%	-0.8%
Waste handling	-4.4%	-2.8%	-0.9%	-5.0%	-2.8%	-0.7%
Utilities	-4.7%	-3.0%	-0.9%	-6.9%	-3.8%	-1.0%
QA/QC	-2.9%	-1.9%	-0.6%	-4.3%	-2.4%	-0.6%
CapEx excluding working capital	-4.2%	-2.7%	-0.8%	-1.6%	-0.9%	-0.2%
Working capital/in-process inventory	-1.6%	-2.2%	-3.1%	-1.7%	-2.5%	-3.2%
Total:	-32%	-22%	-9%	-40%	-24%	-9%
	[−\$485M]	[−\$513M]	[−\$689M]	[−\$2048M]	[−\$2188M]	[−\$3066M]

Difference in present cost relative to batch case is provided in square brackets. Contribution to present cost difference relative to batch is defined by (B.4). *Since the continuous yield is identical to the batch yield, the contribution of KI cost to overall savings is identically zero in all cases.*

and waste handling. Moving towards higher KI price in the table means the KI makes up a higher fraction of the expenses, so percentage savings are reduced. However, the working capital (in-process inventory) savings measured in dollars are increased moving towards higher KI price, since the batch process has ten times more in-process inventory. For a given KI price, the percentage savings due to working capital is similar for high and low loadings. This may initially seem counterintuitive since the high-loading processes require five times more KI, the most expensive raw material. However, the high-loading cases also have about four times greater overall cost than the corresponding low-loading cases, so the percentage savings are similar.

In the low API loading scenarios, the largest savings consistently come from (1st) other raw materials, (2nd) utilities, (3rd) waste handling, (4th) CapEx excluding working capital, (5th) labor and materials handling, (6th) QA/QC, and (7th) other organic reagents. The significance of working capital varies depending on the KI price; it is the 1st, 6th, or 8th contributor to cost savings for KI prices of \$3000/kg, \$500/kg, and \$100/kg respectively.

In the high API loading scenarios, the largest savings consistently come from (1st) other raw materials, (2nd) utilities, (3rd) labor and materials handling, (4th) waste handling, (5th) QA/QC, (6th) other organic reagents, (7th) CapEx excluding working capital. The significance of working capital again depends on the KI price; it is the 1st, 5th, or 7th most significant contributor to cost savings for KI prices of \$3000/kg, \$500/kg, and \$100/kg respectively.

B.5 Discussion

Batch and continuous production of a very large-scale pharmaceutical product produced in dedicated batch or continuous plants was analyzed. This is one of the first market segments in which continuous pharmaceutical manufacturing may be implemented. Overall cost savings of 9 to 40% are predicted if the appropriate process is selected for the API loading/KI price scenario at hand. The novel process with recycling (CM1R) with direct tablet formation is consistently one of the most favorable processes, with the same or slightly more savings than CM1 with roller compaction. Percentage savings are greatest when KI prices are lower. This is because the expenditure for the KI is the same in batch and CM1R, that expenditure is proportional to KI price, and all other expenses except working capital are insensitive to KI price. That is, as KI price approaches infinity, present cost savings approach the working capital savings from the reduced in-process inventory.

The process development costs will tend to be greater for continuous manufacturing processes as opposed to batch processes, because the pharmaceutical industry has less experience with continuous processing and the absence of conventional batches in the highly-regulated industry demands more process understanding and on-line instrumentation (i.e., PAT). This was accounted for as a 10% price premium for continuous processing equipment at the same scale as an equivalent batch process. With the current trend towards smaller continuous processes (e.g., microreactors), however, more process understanding will be obtained at early stages of the process development, making scaleup easier and less expensive over time. Furthermore, the smaller scale required for each unit in a continuous process (due to greater effective utilization time) usually offsets the additional process understanding and control required. Some unit operations are easier to characterize in continuous mode: Dhenge et al [61] claim that continuous granulation processes can be developed more quickly, with associated savings in API material during development. Once the process is operational, labor costs are typically lower as well.

The capital expenditure for the novel direct tablet formation process was based on a vendor quotation, but since it has not been used in the pharmaceutical industry, the equipment cost is subject to more uncertainty than most of the other costs. A typical Wroth factor for the process under consideration in other industries is about 2.0, but the standard value for Other Equipment of 3.5 has been assumed (cf. Table B.3). This assumption allows for cost increases specific to the pharmaceutical industry, and may be unnecessarily high, meaning that the realized cost savings for the direct tablet formation process may be greater than those estimated here. The novel direct tablet formation process should also be more broadly applicable than roller compaction and eliminate other costs related to powder handling that are not considered in this study. In addition to the direct tablet formation approach reported here, we are developing another novel tablet formation process (not yet published) that should have significantly better yields than roller compaction.

Although different aspects of continuous pharmaceutical production have been analyzed [151, 154, 155, 199], no articles have been published comparing the overall economics of batch and continuous pharmaceutical processes producing drug product (tablets) from an organic key intermediate. A presentation at the American Association of Pharmaceutical Scientists meeting on Drug Product Manufacturing claimed 58% CapEx savings and 67% annual OpEx savings for a continuous pharmaceutical processing facility versus a batch processing facility [56]. A study on production of ethanol estimated 57% CapEx savings by shifting the process from batch to continuous mode [57]. A study on production of cell culture media estimated overall cost savings at 34% for switching from batch to continuous production at 100,000 L/yr capacity [90]. A study on production of fine chemicals on dedicated batch vs. continuous equipment found that continuous production is economically favorable at all production levels studied (as low as 200 mtpy) [76]. Thus, this case study of a specific pharmaceutical product found results similar to those found in other industries.

Some areas of continuous processing in pharmaceuticals are well understood whereas others require further study. Microreactors and other continuous-flow reactors have received quite a lot of study [37, 75, 99, 154, 155], however efficient chemistry for the particular product being produced is absolutely crucial. Particularly in the case of high KI price, a 10% yield difference can shift a continuous process from providing cost savings to providing cost increases (see Tables B.11 and B.12). Continuous granulation processes have also been widely studied [61, 73, 92, 93, 118, 168, 200, 208]. On the contrary, studies on continuous crystallization for pharmaceuticals has been lacking until recently

[109]. The recent research is promising: one study found that the purification of an API using a continuous oscilattory baffled crystallizer is much more predictable to scale up than batch crystallization, as well as having a lower residence time: the continuous process took 12 minutes as opposed to 9 hours and 40 minutes using the batch process [109]. The easy scaleup in this type of continuous crystallizer compares favorably with the ten different schemes for scaling up batch crystallization enumerated by Lawton et al [55, 68, 83, 96, 103, 109, 113, 140, 144, 184, 189, 198, 215]. Despite recent promising results on continuous crystallization of pharmaceuticals, appropriate solvents and conditions must be chosen for each specific purification step in each particular manufacturing process. Other effective separations technologies may become more promising under continuous mode as well [149, 207, 217, 218]. Vervaet and Remon [208] wrote a review article on six different methods of continuous granulation. The best-studied method for continuous granulation is extrusion, on which the first papers for pharmaceutical applications were published in 1986 [73] and much subsequent work has been completed [61, 92, 93, 118, 168, 200]. Commercial equipment for creating the final dosage form such as continuous tableting and coating is already available; alternative methods for doing so may prove even more efficient.

Apart from considerations of continuous pharmaceutical manufacturing unit operations, more economic analysis and system-level research is also required. Specifically, analysis of smaller-scale production and considering a multipurpose continuous production line rather than a line dedicated to a single product. For example, how much time is needed to change between products, and how much waste material is generated during startup and shutdown, when the production line has not reached steady state, as well as the economic implications thereof. Plantwide dynamic models are essential for this task [106].

B.6 Conclusion

An integrated cost estimation of the production of a final drug product from a key organic intermediate was performed, using a batch process and four continuous processes. In order to make the analysis applicable to a wider range of products, the analysis was performed with two API loading levels in the final drug product, three prices for the most expensive KI organic feedstock, three continuous API synthesis processes, and two continuous drug product formation processes. The overall cost of production can be reduced most by changing to the continuous process with recycling (CM1R) with the novel direct tablet formation process in all scenarios tested if overall yields for the continuous process meet or exceed those of the batch process; in those two yield scenarios, the savings are 9 to 40% and 19 to 44% respectively. If the CM process with recycling has 10% lower yield than the batch plant, savings can be achieved for all scenarios except the highest KI price. The break-even KI price is \$1700/kg. Again, the maximal savings can be achieved by choosing process CM1R with the novel direct tablet formation process. When combining the economic advantage with more consistent product quality and greater regulatory freedom, continuous manufacturing of pharmaceuticals is a viable way for the pharmaceutical industry to achieve substantial cost savings. Many opportunities for further study exist: developing more efficient chemical routes, separations technologies, final dosage form production, and plantwide modeling are all expected to lead to more economical processes.

B.7 Acknowledgments

This work was funded by Novartis Pharmaceuticals as part of the Novartis-MIT Center for Continuous Manufacturing. The authors thank Kristopher Wilburn (MIT) for vendor price quotations and insight into the batch process and the full Novartis-MIT Center for Continuous Manufacturing team, especially Berthold Schenkel. The authors are grateful to the anonymous reviewers, whose comments led to significant improvements to the paper.

B.8 Nomenclature

API = active pharmaceutical ingredient

BLIC = battery-limits installed cost, US\$

Bx = batch manufacturing process

CapEx = capital expenditures, US\$

CCM = Novartis-MIT Center for Continuous Manufacturing

CM1 = novel continuous manufacturing process

CM1R = novel continuous manufacturing process with recycle

DP = drug product (final dosage form)

DTF = direct tablet formation

FOB = free on board (cost of equipment before delivery), US$

KI = key intermediate

LLE = liquid-liquid extraction

NPV = net present value

OEE = overall equipment effectiveness

OpEx = operating expenditures, US$

PAT = process analytical technology

QA/QC = quality assurance/quality control

RC = roller compaction

Bibliography

[1] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, *A global optimization method, αBB, for general twice-differentiable constrained NLPs II. Implementation and computational results*, Computers & Chemical Engineering, 22 (1998), pp. 1159–1179.

[2] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier, *A global optimization method, αBB, for general twice-differentiable constrained NLPs I. Theoretical advances*, Computers & Chemical Engineering, 22 (1998), pp. 1137–1158.

[3] K.-d. Ahn, Y.-h. Lee, and D.-I. Koo, *Synthesis and polymerization of facile deprotection of polymer side-chain t-BOC groups*, Polymer, 33 (1992), pp. 4851–4856.

[4] G. Alefeld and G. Mayer, *Interval analysis: theory and applications*, Journal of Computational and Applied Mathematics, 121 (2000), pp. 421–464.

[5] M. Amrhein, *Reaction and flow variants/invariants for the analysis of chemical reaction data*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 1998.

[6] R. Aris, *On the Dispersion of a Solute in a Fluid Flowing through a Tube*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 235 (1956), pp. 67–77.

[7] J.-P. Aubin, *Viability Theory*, Birkhauser Boston, Boston, 2009.

[8] M. E. Bailey, V. Kirss, and R. G. Spaunburgh, *Reactivity of Organic Isocyanates*, Industrial & Engineering Chemistry Research, 48 (1956), pp. 794–797.

[9] J. W. Baker, M. M. Davies, and J. Gaunt, *The mechanism of the reaction of aryl isocyanates with alcohols and amines. Part IV. The evidence of infra-red absorption spectra regarding alcohol-amine association in the base-catalysed reaction of phenyl isocyanate with alcohols*, Journal of the Chemical Society, 19 (1949), pp. 24–27.

[10] J. W. Baker and J. Gaunt, *The mechanism of the reaction of aryl isocyanates with alcohols and amines. Part II. The base-catalysed reaction of phenyl isocyanate with alcohols*, Journal of the Chemical Society, 19 (1949), pp. 9–18.

[11] J. W. Baker and J. Gaunt, *The mechanism of the reaction of aryl isocyanates with alcohols and amines. Part III. The "spontaneous" reaction of phenyl isocyanate with various alcohols. Further evidence relating to the anomalous effect of dialkylanilines in the base-catalysed react*, Journal of the Chemical Society, 19 (1949), pp. 19–24.

[12] ——, *The mechanism of the reaction of aryl isocyanates with alcohols and amines. Part V. Kinetic investigations of the reaction between phenylisocyanate and methyl and ethyl alcohols in benzene solution*, Journal of the Chemical Society, 19 (1949), pp. 27–31.

[13] W. Baker and J. B. Holdsworth, *The mechanism of aromatic side-chain reactions with special reference to the polar effects of substituents. Part XIII. Kinetic examination of the reaction of aryl isocyanates with methyl alcohol*, Journal of the Chemical Society, (1947), pp. 713–726.

[14] J. R. Banga, A. A. Alonso, and R. P. Singh, *Stochastic Dynamic Optimization of Batch and Semicontinuous Bioprocesses*, Biotechnology Progress, 13 (1997), pp. 326–335.

[15] I. Bauer, H. G. Bock, S. Körkel, and J. P. Schlöder, *Numerical methods for optimum experimental design in DAE systems*, Journal of Computational and Applied Mathematics, 120 (2000), pp. 1–25.

[16] K. J. Beers, *Numerical Methods for Chemical Engineering*, Cambridge University Press, Cambridge, UK, 2007.

[17] A. Behr, V. Brehme, C. Ewers, H. Grön, T. Kimmel, S. Küppers, and I. Symietz, *New Developments in Chemical Engineering for the Production of Drug Substances*, Engineering in Life Sciences, 4 (2004), pp. 15–24.

[18] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[19] ——, *Introduction to the Mathematical Theory of Control Processes, Vol. 2*, Academic Press, New York, 1971.

[20] R. E. Bellman and R. E. Dreyfus, *Dynamic Programming and Modern Control Theory*, Academic Press, Orlando, Florida, 1977.

[21] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, *Branching and bounds tightening techniques for non-convex MINLP*, Optimization Methods and Software, 24 (2009), pp. 597–634.

[22] C. Bendtsen and O. Stauning, *FADBAD, a flexible C++ package for automatic differentiation*, tech. report, Technical University of Denmark, Lyngby, Denmark, 1996.

[23] H. Berthiaux, K. Marikh, and C. Gatumel, *Continuous mixing of powder mixtures with pharmaceutical process constraints*, Chemical Engineering and Processing, 47 (2008), pp. 2315–2322.

[24] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 2nd ed., 1999.

[25] ——, *Dynamic Programming and Optimal Control, Vol. 1*, Athena Scientific, Belmont, MA, 2005.

[26] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM, Philadelphia, 2010.

[27] L. T. Biegler, *Solution of dynamic optimization problems by successive quadratic*

programming and orthogonal collocation, Computers & Chemical Engineering, 8 (1984), pp. 243–248.

[28] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms, and Applications to Chermical Processes*, SIAM, Philadelphia, PA, 2010.

[29] L. T. Biegler, A. M. Cervantes, and A. Wächter, *Advances in simultaneous strategies for dynamic process optimization*, Chemical Engineering Science, 57 (2002), pp. 575–593.

[30] J. Björnberg and M. Diehl, *Approximate robust dynamic programming and robustly stable MPC*, Automatica, 42 (2006), pp. 777–782.

[31] R. V. Boltyanskii, V. G., Gamkrelidze and L. S. Pontryagin, *On the Theory of Optimal Processes (in Russian)*, Doklady Akademii Nauk SSSR, 110 (1956), pp. 7–10.

[32] A. Bompadre and A. Mitsos, *Convergence rate of McCormick relaxations*, Journal of Global Optimization, 52 (2012), pp. 1–28.

[33] A. Bompadre, A. Mitsos, and B. Chachuat, *Convergence analysis of Taylor models and McCormick-Taylor models*, Journal of Global Optimization, 57 (2013), pp. 75–114.

[34] G. E. P. Box and W. J. Hill, *Discrimination among mechanistic models*, Technometrics, 9 (1967), pp. 57–71.

[35] G. E. P. Box and W. G. Hunter, *The Experimental Study of Physical Mechanisms*, Technometrics, 7 (1965), pp. 23–42.

[36] G. E. P. Box and H. L. Lucas, *Design of Experiments in Non-Linear Situations*, Biometrika, 46 (1959), pp. 77–90.

[37] C. E. Brocklehurst, H. Lehmann, and L. L. Vecchia, *Nitration Chemistry in Continuous Flow using Fuming Nitric Acid in a Commercially Available Flow Reactor*, Organic Process Research & Development, 15 (2011), pp. 1447–1453.

[38] H. Brönnimann, G. Melquiond, and S. Pion, *The design of the Boost interval arithmetic library*, Theoretical Computer Science, 351 (2006), pp. 111–118.

[39] D. L. Browne, M. Baumann, B. H. Harji, I. R. Baxendale, and S. V. Ley, *A New Enabling Technology for Convenient Laboratory Scale Continuous Flow Processing at Low Temperatures*, Organic Letters, 13 (2011), pp. 3312–3315.

[40] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control*, Taylor & Francis, Bristol, PA, 1975.

[41] G. Buzzi-Ferraris and P. Forzatti, *A new sequential experimental design procedure for discriminating among rival models*, Chemical Engineering Science, 38 (1983), pp. 225–232.

[42] ——, *Sequential experimental design for model discrimination in the case of multiple responses*, Chemical Engineering Science, 39 (1984), pp. 81–85.

[43] G. Buzzi-Ferraris and F. Manenti, *Kinetic models analysis*, Chemical Engineering Science, 64 (2009), pp. 1061–1074.

[44] E. F. Carrasco and J. R. Banga, *Dynamic optimization of batch reactors using adaptive stochastic algorithms*, Industrial & Engineering Chemistry Research, 36 (1997), pp. 2252–2261.

[45] A. Cervantes and L. T. Biegler, *Large-Scale DAE Optimization Using a Simultaneous NLP Formulation*, AIChE Journal, 44 (1998), pp. 1038–1050.

[46] A. M. Cervantes, A. Wächter, R. H. Tütüncü, and L. T. Biegler, *A reduced space interior point strategy for optimization of differential algebraic systems*, Computers and Chemical Engineering, 24 (2000), pp. 39–51.

[47] B. Chachuat, *RESEARCH-b.chachuat http://www3.imperial.ac.uk/people/b.chachuat/*, tech. report, http://www3.imperial.ac.uk/people/b.chachuat/research, 2011.

[48] B. Chachuat, P. I. Barton, and A. B. Singer, *Global methods for dynamic optimization and mixed-integer dynamic optimization*, Industrial & Engineering Chemistry Research, 45 (2006), pp. 8373–8392.

[49] M.-C. Chang and S.-A. Chen, *Kinetics and mechanism of urethane reactions: Phenyl isocyanate-alcohol systems*, Journal of Polymer Science, Part A: Polymer Chemistry, 25 (1987), pp. 2543–2559.

[50] Y. Chen, Y. Zhao, M. Han, C. Ye, M. Dang, and G. Chen, *Safe, efficient and selective synthesis of dinitro herbicides via a multifunctional continuous-flow microreactor: one-step dinitration with nitric acid as agent*, Green Chemistry, 15 (2013), pp. 91–94.

[51] M. Cizniar, M. Podmajerský, T. Hirmajer, M. Fikar, and A. M. Latifi, *Global optimization for parameter estimation of differential-algebraic systems*, Chemical Papers, 63 (2009), pp. 274–283.

[52] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, 1990.

[53] J. R. Couper, *Process Engineering Economics*, CRC Press, New York, 2003.

[54] J. R. Couper, D. W. Hertz, and F. L. Smith, *Process Economics*, in Perry's Chemical Engineers' Handbook, D. W. Green and R. H. Perry, eds., McGraw-Hill, New York, 8th ed., 2008, ch. 8, pp. 9–1 – 9–56.

[55] J. R. Couper, W. R. Penney, and J. R. Fair, *Chemical Process Equipment: Selection and Design*, Elsevier, Amsterdam, Boston, 2004.

[56] T. F. Crosby, *Enhanced Capital Productivity Through Continuous Processing*, American Association of Pharmaceutical Scientists conference on Drug Product Manufacturing, (2010).

[57] G. R. Cysewski and C. R. Wilke, *Process design and economic studies of alternative fermentation methods for the production of ethanol*, Biotechnology and Bioengineering, 20 (1978), pp. 1421–1444.

[58] S. A. Dadebo and K. B. Mcauley, *Dynamic optimization of constrained chemical engineering problems using dynamic programming*, Computers & Chemical Engineering, 19 (1995), pp. 513–525.

[59] G. Dahlquist, *Stability and error bounds in the numerical integration of ordinary differential equations*, PhD thesis, University of Stockholm, 1958.

[60] W. M. Deen, *Analysis of Transport Phenomena*, Oxford University Press, New York, 1998.

[61] R. Dhenge, R. Fyles, J. Cartwright, D. G. Doughty, M. J. Hounslow, and A. D. Salman, *Twin screw wet granulation: Granule properties*, Chemical Engineering Journal, 164 (2010), pp. 322–329.

[62] K. Du and R. B. Kearfott, *The cluster problem in multivariate global optimization*, Journal of Global Optimization, 5 (1994), pp. 253–265.

[63] E. Dyer, H. A. Taylor, S. J. Mason, and J. Samson, *The rates of reaction of isocyanates with alcohols. I. Phenyl isocyanate with 1- and 2-butanol*, Journal of the American Chemical Society, 71 (1949), pp. 4106–4109.

[64] G. I. Egorov and A. M. Kolker, *The Thermal Properties of Water-N, N-Dimethylformamide Solutions at 278-323.15 K and 0.1-100 MPa*, Russian Journal of Physical Chemistry A, 82 (2008), pp. 2058–2064.

[65] T. G. W. Epperly and E. N. Pistikopoulos, *A Reduced Space Branch and Bound Algorithm for Global Optimization*, Journal of Global Optimization, (1997), pp. 287–311.

[66] W. R. Esposito and C. A. Floudas, *Deterministic global optimization in nonlinear optimal control problems*, Journal of Global Optimization, (2000), pp. 97–126.

[67] W. R. Esposito and C. A. Floudas, *Global optimization for the parameter estimation of differential-algebraic systems*, Industrial & Engineering Chemistry Research, 39 (2000), pp. 1291–1310.

[68] J. J. Evangelista, S. Katz, and R. Shinnar, *Scale-up criteria for stirred tank reactors*, AIChE Journal, 15 (2004), pp. 843–853.

[69] W. F. Feehery, J. E. Tolsma, and P. I. Barton, *Efficient sensitivity analysis of large-scale differential-algebraic systems*, Applied Numerical Mathematics, 25 (1997), pp. 41–54.

[70] G. Franceschini and S. Macchietto, *Model-based design of experiments for parameter precision: State of the art*, Chemical Engineering Science, 63 (2008), pp. 4846–4872.

[71] S. Galán, W. F. Feehery, and P. I. Barton, *Parametric sensitivity functions for hybrid discrete/continuous systems*, Applied Numerical Mathematics, 31 (1999), pp. 17–47.

[72] F. Galvanin, M. Barolo, and F. Bezzo, *Online model-based redesign of experiments for parameter estimation in dynamic systems*, Industrial & Engineering Chemistry Research, 48 (2009), pp. 4415–4427.

[73] M. Gamlen and C. Eardley, *Continuous extrusion using a Baker Perkins MP50 (Multipurpose) extruder*, Drug Development and Industrial Pharmacy, 12 (1986), pp. 1701–1713.

[74] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT : An SQP Algorithm for Large-Scale Constrained Optimization*, SIAM Review, 47 (2005), pp. 99–131.

[75] T. N. Glasnov and C. O. Kappe, *Toward a Continuous-Flow Synthesis of Boscalid*, Advanced Synthesis & Catalysis, 352 (2010), pp. 3089–3097.

[76] A. Gorsek and P. Glavic, *Design of Batch Versus Continuous Processes Part III: Extended Analysis of Cost Parameters*, Chemical Engineering Research & Design, 78 (2000), pp. 231–244.

[77] A. Griewank, *Automatic directional differentiation of nonsmooth composite functions*, in Recent Developments in Optimization, French-German Conference on Optimization, Dijon, 1994.

[78] L. Grüne and W. Semmler, *Using dynamic programming with adaptive grid scheme for optimal control problems in economics*, Journal of Economic Dynamics and Control, 28 (2004), pp. 2427–2456.

[79] J. W. Hagood and B. S. Thomson, *Recovering a Function from a Dini Derivative*, The American Mathematical Monthly, 113 (2006), pp. 34–46.

[80] E. Hairer, S. P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I*, Springer-Verlag, Berlin, 1993.

[81] G. Harrison, *Dynamic models with uncertain parameters*, in Proceedings of the First International Conference on Mathematical Modeling, X. Avula, ed., vol. 1, University of Missouri Rolla, 1977, pp. 295–304.

[82] P. D. H. Hill, *A review of experimental design procedures for regression model discrimination*, Technometrics, 20 (1978), pp. 15–21.

[83] F. W. J. M. M. Hoeks, L. A. Boon, F. Studer, M. O. Wolff, F. van der Schot, P. Vrabel, R. G. J. M. van der Lans, W. Bujalski, A. Manelius, G. Blomsten, S. Hjorth, G. Prada, K. C. A. M. Luyben, and A. W. Nienow, *Scale up of stirring as foam disruption (SAFD) to industrial scale*, Journal of Industrial Microbiology and Biotechnology, 30 (2003), pp. 118–128.

[84] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer, Berlin, third ed., 1996.

[85] B. Houska and B. Chachuat, *Branch-and-Lift Algorithm for Deterministic Global Optimization in Nonlinear Optimal Control*, Journal of Optimization Theory and Applications, in press (2013).

[86] B. Houska, M. E. Villanueva, and B. Chachuat, *A Validated Integration Algorithm for Nonlinear ODEs using Taylor Models and Ellipsoidal Calculus*, in 52nd IEEE Conference on Decision and Control, 2013, pp. 484–489.

[87] W. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer-Verlag, Berlin, 2003.

[88] W. G. Hunter and A. M. Reiner, *Designs for Discriminating between Two Rival Models*, Technometrics, 7 (1965), pp. 307–323.

[89] International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use, *Pharmaceutical Development Q8(R2)*, Tech. Report August, International Conference on Harmonisation of Technology, 2009.

[90] D. W. Jayme, J. M. Kubiak, T. A. Battistoni, and D. J. Cady, *Continuous, high capacity reconstitution of nutrient media from concentrated intermediates*, Cytotechnology, 22 (1996), pp. 255–261.

[91] S. Kameswaran and L. T. Biegler, *Simultaneous dynamic optimization strategies: Recent advances and challenges*, Computers & Chemical Engineering, 30 (2006), pp. 1560–1575.

[92] E. Keleb, A. Vermeire, C. Vervaet, and J. Remon, *Continuous twin screw extrusion for the wet granulation of lactose*, International Journal of Pharmaceutics, 239 (2002), pp. 69–80.

[93] ——, *Twin screw granulation as a simple and efficient tool for continuous wet granulation*, International Journal of Pharmaceutics, 273 (2004), pp. 183–194.

[94] H. K. Khalil, *Nonlinear Systems*, Prentice-Hall, Upper Saddle River, NJ, 3rd ed., 2002.

[95] K. A. Khan and P. I. Barton, *Sliding modes of solutions of nonsmooth ordinary differential equations*, In preparation, (2014).

[96] J. K. Kim, C. K. Kim, and J. Kawasaki, *A Scale up of Stirred Tank Contactors for the Liquid Membrane Permeation of Hydrocarbons.*, Separation Science and Technology, 36 (2001), pp. 3585–3598.

[97] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by Simulated Annealing*, Science, 220 (1983), pp. 671–680.

[98] O. Knüppel, *PROFIL / BIAS—A Fast Interval Library*, Computing, 53 (1994), pp. 277–287.

[99] N. Kockmann, M. Gottsponer, B. Zimmermann, and D. M. Roberge, *Enabling continuous-flow chemistry in microstructured devices for pharmaceutical and fine-chemical production.*, Chemistry: A European Journal, 14 (2008), pp. 7470–7.

[100] N. Kockmann and D. M. Roberge, *Harsh Reaction Conditions in Continuous-Flow Microreactors for Pharmaceutical Production*, Chemical Engineering & Technology, 32 (2009), pp. 1682–1694.

[101] S. Körkel, *Numerische Methoden für Optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen*, PhD thesis, Universität Heidelberg, 2002.

[102] S. Korkel, J. P. Schloder, E. Kostina, and H. G. Bock, *Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes*, Optimization Methods and Software, 19 (2004), pp. 327–338.

[103] M. Kraume and P. Zehner, *Concept for scale-up of solids suspension in stirred tanks*, Can. J. Chem. Eng, 80 (2002), pp. 674–681.

[104] A. A. Kulkarni, V. S. Kalyani, R. A. Joshi, and R. R. Joshi, *Continuous Flow Nitration of Benzaldehyde*, Organic Process Research & Development, 13 (2009), pp. 999–1002.

[105] R. Lakerveld, B. Benyahia, R. D. Braatz, and P. I. Barton, *Model-Based Design of a Plant-Wide Control Strategy for a Continuous Pharmaceutical Plant*, AIChE Journal, 59 (2013), pp. 3671–3685.

[106] R. Lakerveld, R. D. Braatz, and P. I. Barton, *A Plant-Wide Control Strategy for Continuous Pharmaceutical Manufacturing*, AIChE Annual Meeting, (2010).

[107] T. LaPorte and C. Wang, *Continuous processes for the production of pharmaceutical intermediates and active pharmaceutical ingredients*, Current Opinion in Drug Discovery & Development, 10 (2007), pp. 738–745.

[108] E. L. Lawler and D. E. Wood, *Branch-and-Bound Methods: A Survey*, Operations Research, 14 (1966), pp. 699–719.

[109] S. Lawton, G. Steele, P. Shering, L. Zhao, I. Laird, and X.-W. Ni, *Continuous Crystallization of Pharmaceuticals Using a Continuous Oscillatory Baffled Crystallizer*, Organic Process Research & Development, 13 (2009), pp. 1357–1363.

[110] D. B. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder, *An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: theoretical aspects*, Computers and Chemical Engineering, 27 (2003), pp. 157–166.

[111] M. Lerch, G. Tischler, J. W. von Gudenberg, W. Hofschuster, and W. K. Amer, *FILIB++, A Fast Interval Library Supporting Containment Computations*, ACM Transactions on Mathematical Software, 32 (2006), pp. 299–324.

[112] H. Leuenberger, *New trends in the production of pharmaceutical granules: batch versus continuous processing*, European Journal of Pharmaceutics and Biopharmaceutics, 52 (2001), pp. 289–296.

[113] M. Li, G. White, D. Wilkinson, and K. J. Roberts, *Scale up study of retreat curve impeller stirred tanks using LDA measurements and CFD simulation*, Chemical Engineering Journal, 108 (2005), pp. 81–90.

[114] Y. Lin and M. A. Stadtherr, *Deterministic global optimization for parameter estimation of dynamic systems*, Industrial and Engineering Chemistry Research, 45 (2006), pp. 8438–8448.

[115] Y. Lin and M. A. Stadtherr, *Validated solution of ODEs with parametric uncertainties*, 16th European Symposium on Computer Aided Process, (2006).

[116] Y. Lin and M. A. Stadtherr, *Deterministic Global Optimization of Nonlinear Dynamic Systems*, AIChE Journal, 53 (2007), pp. 866–875.

[117] Y. Lin and M. A. Stadtherr, *Validated solutions of initial value problems for parametric ODEs*, Applied Numerical Mathematics, 57 (2007), pp. 1145–1162.

[118] N. Lindberg, C. Tufvesson, P. Holm, and L. Olbjer, *Extrusion of an effervescent granulation with twin screw extruder, Baker Perkins MPF50D. Influence of intragranular porosity and liquid saturation.*, Drug Development and Industrial Pharmacy, 14 (1988), pp. 1791–1798.

[119] J. S. Logsdon and L. T. Biegler, *Decomposition strategies for large-scale dynamic optimization problems*, Chemical Engineering Science, 47 (1992), pp. 851–864.

[120] R. Luus, *Optimal control by dynamic programming using systematic reduction in grid size*, International Journal of Control, 51 (1990), pp. 995–1013.

[121] R. Luus, J. Dittrich, and F. J. Keil, *Multiplicity of solutions in the optimization of a bifunctional catalyst blend in a tubular reactor*, The Canadian Journal of Chemical Engineering, 70 (1992), pp. 780–785.

[122] P. Majer and R. S. Randad, *A Safe and Efficient Method for Preparation of N,N'-Unsymmetrically Disubstituted Ureas Utilizing Triphosgene*, J. Org. Chem., 59 (1994), pp. 1937–1938.

[123] T. Maly and L. R. Petzold, *Numerical methods and software for sensitivity analysis of differential-algebraic systems*, Applied Numerical Mathematics, 20 (1996), pp. 57–79.

[124] R. A. Maurya, C. P. Park, J. H. Lee, and D.-P. Kim, *Continuous in situ generation, separation, and reaction of diazomethane in a dual-channel microreactor*, Angewandte Chemie (International ed. in English), 50 (2011), pp. 5952–5955.

[125] G. P. McCormick, *Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems*, Mathematical Programming, 10 (1976), pp. 147–175.

[126] P. McKenzie, S. Kiang, J. Tom, A. E. Rubin, and M. Futran, *Can Pharmaceutical Process Development Become High Tech?*, AIChE Journal, 52 (2006), pp. 3990–3994.

[127] J. P. Mcmullen, *Automated Microreactor System for Reaction Development and Online Optimization of Chemical Processes*, PhD thesis, Massachusetts Institute of Technology, 2010.

[128] J. P. McMullen and K. F. Jensen, *Rapid Determination of Reaction Kinetics with an Automated Microfluidic System*, Organic Process Research & Development, 15 (2011), pp. 398–407.

[129] F. Messine, *Deterministic global optimization using interval constraint propagation techniques*, RAIRO Operations Research, 38 (2004), pp. 277–293.

[130] A. Mitsos, B. Chachuat, and P. I. Barton, *McCormick-based relaxations of algorithms*, SIAM Journal on Optimization, 20 (2009), pp. 573–601.

[131] C. G. Moles, P. Mendes, and J. R. Banga, *Parameter estimation in biochemical pathways: a comparison of global optimization methods*, Genome Research, 13 (2003), pp. 2467–2474.

[132] J. S. Moore, *Kinetic Modeling and Automated Optimization in Microreactor Systems*, PhD thesis, Massachusetts Institute of Technology, 2013.

[133] J. S. Moore and K. F. Jensen, *"Batch" Kinetics in Flow: Online IR Analysis and Continuous Control*, Angewandte Chemie (International ed. in English), 53 (2014), pp. 470–473.

[134] R. E. Moore, *Interval arithmetic and automatic error analysis in digital computing*, PhD thesis, Stanford University, 1962.

[135] ——, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[136] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[137] S. Mozharov, A. Nordon, D. Littlejohn, C. Wiles, P. Watts, P. Dallin, and J. M. Girkin, *Improved method for kinetic studies in microreactors using flow*

manipulation and noninvasive Raman spectrometry, Journal of the American Chemical Society, 133 (2011), pp. 3601–3608.

[138] A. Nagaki, Y. Takahashi, S. Yamada, C. Matsuo, S. Haraki, Y. Moriwaki, S. Kim, and J.-i. Yoshida, *Generation and Reactions of Vinyllithiums Using Flow Microreactor Systems*, Journal of Flow Chemistry, 2 (2012), pp. 70–72.

[139] A. Nagaki, S. Yamada, M. Doi, Y. Tomida, N. Takabayashi, and J.-i. Yoshida, *Flow microreactor synthesis of disubstituted pyridines from dibromopyridines via Br/Li exchange without using cryogenic conditions*, Green Chemistry, 13 (2011), pp. 1110–1113.

[140] E. Nauman, *Chemical Reactor Design, Optimization, and Scaleup*, McGraw-Hill Professional, New York, 2002.

[141] A. K. W. Navarro and V. S. Vassiliadis, *Computer Algebra Systems Coming of Age: Dynamic Simulation and Optimisation of DAE systems in Mathematica*, Computers & Chemical Engineering, In press (2013).

[142] A. Neumaier, *Taylor Forms Use and Limits*, Reliable Computing, (2003), pp. 43–79.

[143] ——, *Complete search in continuous global optimization and constraint satisfaction*, Acta Numerica, 13 (2004), pp. 271–369.

[144] A. W. Nienow, M. F. Edwards, and N. Harnby, *Mixing in the Process Industries*, Butterworth-Heinemann, Woburn, MA, 2nd ed., 1997.

[145] N. Oi and H. Kitahara, *High-performance liquid chromatographic separation of chiral alcohols on chiral stationary phases*, Journal of Chromatography, 265 (1983), pp. 117–120.

[146] S. Oi, H. Ono, H. Tanaka, Y. Matsuzaka, and S. Miyano, *Investigation on the chiral discrimination mechanism using an axially asymmetric binaphthalene-based stationary phase for high-performance liquid chromatography*, Journal of Chromatography A, 659 (1994), pp. 75–86.

[147] S. Oi, M. Shijo, H. Tanaka, and S. Miyano, *Chiral stationary phases consisting of axially dissymmetric 2'-substituted-1, 1'-binaphthyl-2-carboxylic acids bonded to silica gel for high-performance liquid chromatographic separation of enantiomers*, Journal of Chromatography, 645 (1993), pp. 17–28.

[148] I. Papamichail and C. S. Adjiman, *A rigorous global optimization algorithm for problems with ordinary differential equations*, Journal of Global Optimization, 24 (2002), pp. 1–33.

[149] S. Peper, M. Lübbert, M. Johannsen, and G. Brunner, *Separation of ibuprofen enantiomers by supercritical fluid simulated moving bed chromatography*, Separation Science and Technology, 37 (2002), pp. 2545–2566.

[150] D. P. Petrides, A. Koulouris, and P. T. Lagonikos, *The Role of Process Simulation in Pharmaceutical Process Development and Product Commercialization*, Pharmaceutical Engineering, 22 (2002), pp. 1–8.

[151] K. Plumb, *Continuous Processing in the Pharmaceutical Industry: Changing the Mind Set*, Chemical Engineering Research & Design, 83 (2005), pp. 730–738.

[152] L. S. Pontryagin, *The Mathematical Theory of Optimal Processes*, Interscience, New York, 1962.

[153] B. J. Reizman, *Personal communication*, 2013.

[154] D. M. Roberge, L. Ducry, N. Bieler, P. Cretton, and B. Zimmermann, *Microreactor Technology: A Revolution for the Fine Chemical and Pharmaceutical Industries?*, Chemical Engineering & Technology, 28 (2005), pp. 318–323.

[155] D. M. Roberge, B. Zimmermann, F. Rainone, M. Gottsponer, M. Eyholzer, and N. Kockmann, *Microreactor Technology and Continuous Processes in the Fine Chemical and Pharmaceutical Industry: Is the Revolution Underway?*, Organic Process Research & Development, 12 (2008), pp. 905–910.

[156] M. Rodriguez-Fernandez, J. A. Egea, and J. R. Banga, *Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems.*, BMC Bioinformatics, 7 (2006), p. 483.

[157] O. Rosen and R. Luus, *Global optimization approach to nonlinear optimal control*, Journal of Optimization Theory and Applications, 73 (1992), pp. 547–562.

[158] S. M. Rump, *INTLAB - INTerval LABoratory*, in Developments in Reliable Computing, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, pp. 77–104.

[159] H. S. Ryoo and N. V. Sahinidis, *Global optimization of nonconvex NLPs and MINLPs with applications in process design*, Computers & Chemical Engineering, 19 (1995), pp. 551–566.

[160] ——, *A branch-and-reduce approach to global optimization*, Journal of Global Optimization, 8 (1996), pp. 107–138.

[161] N. V. Sahinidis, *BARON : A General Purpose Global Optimization Software Package*, Journal of Global Optimization, 8 (1996), pp. 201–205.

[162] A. M. Sahlodin, *Global Optimization of Dynamic Process Systems Using Complete Search Methods*, PhD thesis, McMaster University, 2013.

[163] A. M. Sahlodin and B. Chachuat, *Convex/concave relaxations of parametric ODEs using Taylor models*, Computers & Chemical Engineering, 35 (2011), pp. 844–857.

[164] A. M. Sahlodin and B. Chachuat, *Discretize-then-relax approach for convex/concave relaxations of the solutions of parametric ODEs*, Applied Numerical Mathematics, 61 (2011), pp. 803–820.

[165] S. D. Schaber, D. I. Gerogiorgis, R. Ramachandran, J. M. B. Evans, P. I. Barton, and B. L. Trout, *Economic Analysis of Integrated Continuous and Batch Pharmaceutical Manufacturing: A Case Study*, Industrial & Engineering Chemistry Research, 50 (2011), pp. 10083–10092.

[166] A. Schöbel and D. Scholz, *The theoretical and empirical rate of convergence for geometric branch-and-bound methods*, Journal of Global Optimization, 48 (2010), pp. 473–495.

[167] D. Scholz, *Theoretical rate of convergence for interval inclusion functions*, Journal of Global Optimization, 53 (2012), pp. 749–767.

[168] R. Schroeder and K. Steffens, *Ein neuartiges system für die kontinuierliche Feuchtgranulierung.*, Pharmazeutische Industrie, 64 (2002), pp. 283–288.

[169] K. Schwetlick and R. Noack, *Kinetics and Catalysis of Consecutive Isocyanate Reactions. Formation of Carbamates, Allophanates and lsocyanurates*, Journal of the Chemical Society, Perkin Transactions, 2 (1995).

[170] J. K. Scott, *Reachability Analysis and Deterministic Global Optimization of Differential-Algebraic Systems*, PhD thesis, Massachusetts Institute of Technology, 2012.

[171] J. K. Scott and P. I. Barton, *Tight, efficient bounds on the solutions of chemical kinetics models*, Computers & Chemical Engineering, 34 (2010), pp. 717–731.

[172] ——, *Bounds on the reachable sets of nonlinear control systems*, Automatica, 49 (2013), pp. 93–100.

[173] ——, *Convex and Concave Relaxations for the Parametric Solutions of Semi-explicit Index-One Differential-Algebraic Equations*, Journal of Optimization Theory and Applications, 156 (2013), pp. 617–649.

[174] ——, *Improved relaxations for the parametric solutions of ODEs using differential inequalities*, Journal of Global Optimization, 57 (2013), pp. 143–176.

[175] ——, *Interval bounds on the solutions of semi-explicit index-one DAEs. Part 1: analysis*, Numerische Mathematik, 125 (2013), pp. 1–25.

[176] ——, *Interval bounds on the solutions of semi-explicit index-one DAEs. Part 2: computation*, Numerische Mathematik, 125 (2013), pp. 27–60.

[177] J. K. Scott, B. Chachuat, and P. I. Barton, *Nonlinear convex and concave relaxations for the solutions of parametric ODEs*, Optimal Control Applications and Methods, 34 (2013), pp. 145–163.

[178] J. K. Scott, M. D. Stuber, and P. I. Barton, *Generalized McCormick relaxations*, Journal of Global Optimization, 51 (2011), pp. 569–606.

[179] B. Semporé and J. Bézard, *Enantiomer separation by chiral-phase liquid chromatography of urethane derivatives of natural diacylglycerols previously fractionated by reversed-phase liquid chromatography*, Journal of Chromatography, 557 (1991), pp. 227–240.

[180] B. G. Sempore and J. A. Bezard, *Analysis and fractionation of natural source diacylglycerols as urethane derivatives by reversed-phase high-performance liquid chromatography*, Journal of Chromatography, 547 (1991), pp. 89–103.

[181] J. P. Shectman and N. V. Sahinidis, *A Finite Algorithm for Global Minimization of Separable Concave Programs*, Journal of Global Optimization, 12 (1998), pp. 1–35.

[182] T. Shioiri, K. Ninomiya, and S.-i. Yamada, *Diphenylphosphoryl Azide. A New Convenient Reagent for a Modified Curtius Reaction and for the Peptide Synthesis*, Journal of the American Chemical Society, 94 (1972), pp. 6203–6205.

[183] T. Shioiri and S.-i. Yamada, *Diphenyl Phosphorazidate*, Organic Syntheses, 62 (1984), pp. 187–188.

[184] V. B. Shukla, U. Parasu Veera, P. R. Kulkarni, and A. B. Pandit, *Scaleup of biotransformation process in stirred tank reactor using dual impeller bioreactor*, Biochemical Engineering Journal, 8 (2001), pp. 19–29.

[185] A. B. Singer, *Global Dynamic Optimization*, PhD thesis, Massachusetts Institute of Technology, 2004.

[186] A. B. Singer and P. I. Barton, *Bounding the solutions of parameter dependent nonlinear ordinary differential equations*, SIAM Journal on Scientific Computing, 27 (2006), p. 2167.

[187] ——, *Global optimization with nonlinear ordinary differential equations*, Journal of Global Optimization, (2006), pp. 159–190.

[188] A. B. Singer, J. W. Taylor, P. I. Barton, and W. H. Green, *Global dynamic optimization for parameter estimation in chemical kinetics*, Journal of Physical Chemistry A, 110 (2006), pp. 971–976.

[189] G. W. Smith, L. L. Tavlarides, and J. Placek, *Turbulent flow in stirred tanks: scale-up computations for vessel hydrodynamics*, Chemical Engineering Communications, 93 (1990), pp. 49–73.

[190] G. Söderlind, *The logarithmic norm. History and modern theory*, BIT Numerical Mathematics, 46 (2006), pp. 631–652.

[191] D. A. Spielman and S.-H. Teng, *Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time*, Journal of the ACM, 51 (2004), pp. 385–463.

[192] R. Storn and K. Price, *Differential Evolution — A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*, Journal of Global Optimization, 11 (1997), pp. 341–359.

[193] T. Takagi, N. Aoyanagi, K. Nishimura, Y. Ando, and T. Ota, *Enantiomer separations of secondary alkanols with little asymmetry by high-performance liquid chromatography on chiral columns*, Journal of Chromatography, 629 (1993), pp. 385–388.

[194] M. Tawarmalani and N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*, Kluwer Academic Publishers, Dordrecht, 2002.

[195] M. Tawarmalani and N. V. Sahinidis, *Global optimization of mixed-integer nonlinear programs: A theoretical and computational study*, Mathematical Programming, Ser. A, 99 (2004), pp. 563–591.

[196] G. Taylor, *Dispersion of Soluble Matter in Solvent Flowing Slowly through a Tube*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 219 (1953), pp. 186–203.

[197] K. L. Teo, C. J. Goh, and K. H. Wong, *A unified computational approach to optimal control problems*, Longman Scientific & Technical, Essex, England, 1991.

[198] D. Thoenes, *Chemical Reactor Development: From Laboratory Synthesis to Industrial Production*, Kluwer Academic Publishers, Dordrecht, Boston, 1994.

[199] H. Thomas, *The Reality of Continuous Processing*, Manufacturing Chemist, (2005).

[200] M. Thompson and J. Sun, *Wet granulation in a twin-screw extruder: Implications of screw design*, Journal of Pharmaceutical Sciences, 99 (2010), pp. 2090–2103.

[201] I.-B. Tjoa and L. T. Biegler, *Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems*, Industrial & Engineering Chemistry Research, 30 (1991), pp. 376–385.

[202] T. Togkalidou, M. Fujiwara, S. Patel, and R. D. Braatz, *Solute concentration prediction using chemometrics and ATR-FTIR spectroscopy*, Journal of Crystal Growth, 231 (2001), pp. 534–543.

[203] F. Trachsel, B. Tidona, S. Desportes, and P. Rudolf von Rohr, *Solid catalyzed hydrogenation in a Si/glass microreactor using supercritical CO2 as the reaction solvent*, The Journal of Supercritical Fluids, 48 (2009), pp. 146–153.

[204] G. D. Ulrich, *A Guide to Chemical Engineering Process Design*, Wiley, New York, 1984.

[205] O. Vandenabeele-Trambouze, L. Mion, L. Garrelly, and A. Commeyras, *Reactivity of organic isocyanates with nucleophilic compounds: amines; alcohols; thiols; oximes; and phenols in dilute organic solutions*, Advances in Environmental Research (Oxford, United Kingdom), 6 (2001), pp. 45–55.

[206] V. Vassiliadis, R. Sargent, and C. Pantelides, *Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints*, Industrial & Engineering Chemistry Research, 33 (1994), pp. 2111–2122.

[207] C. Vemavarapu, M. J. Mollan, M. Lodaya, and T. E. Needham, *Design and process aspects of laboratory scale SCF particle formation systems*, International Journal of Pharmaceutics, 292 (2005), pp. 1–16.

[208] C. Vervaet and J. Remon, *Continuous granulation in the pharmaceutical industry*, Chemical Engineering Science, 60 (2005), pp. 3949–3957.

[209] O. von Stryk and R. Bulirsch, *Direct and indirect methods for trajectory optimization*, Annals of Operations Research, 37 (1992), pp. 357–373.

[210] E. Walter, ed., *Identifiability of Parametric Models*, Pergamon, London, 1987.

[211] E. Walter and L. Pronzato, *Qualitative and quantitative experiment design for phenomenological models—A survey*, Automatica, 26 (1990), pp. 195–213.

[212] W. Walter, *Differential and Integral Inequalities*, Springer, Berlin, 1970.

[213] G. Wang, C. Li, J. Li, and X. Jia, *Catalyst-free water-mediated N-Boc deprotection*, Tetrahedron Letters, 50 (2009), pp. 1438–1440.

[214] A. Wechsung, S. D. Schaber, and P. I. Barton, *The cluster problem revisited*, Journal of Global Optimization, 58 (2014), pp. 429–438.

[215] M. J. Whitton and A. W. Nienow, *Scale up correlations for gas holdup and mass transfer coefficients in stirred tank reactors*, in Proceedings of 3rd International Conference on Bioreactor and Bioprocess Fluid Dynamics, A. Nienow, ed., Mechanical Engineering Publications Ltd., Cambridge, UK, 1993.

[216] D. A. Wicks and Z. W. Wicks, *Blocked isocyanates III: Part A. Mechanisms and chemistry*, Progress in Organic Coatings, 36 (1999), pp. 148–172.

[217] M. Wind, P. Hoffmann, H. Wagner, and W. Thormann, *Chiral capillary electrophoresis as predictor for separation of drug enantiomers in continuous flow zone electrophoresis.*, Journal of Chromatography A, 895 (2000), pp. 51–65.

[218] T. Yoon, B. Chung, and I. Kim, *A novel design of simulated moving bed (SMB) chromatography for separation of ketoprofen enantiomer*, Biotechnol. Bioprocess Eng., 9 (2004), pp. 285–291.

[219] Z. Yu, Y. Lv, C. Yu, and W. Su, *A High-Output, Continuous Selective and Heterogeneous Nitration of p-Difluorobenzene*, Organic Process Research & Development, 17 (2013), pp. 438–442.

[220] Z.-q. Yu, Y.-w. Lv, C.-m. Yu, and W.-k. Su, *Continuous flow reactor for Balz-Schiemann reaction: a new procedure for the preparation of aromatic fluorides*, Tetrahedron Letters, 54 (2013), pp. 1261–1263.

[221] M. Yunt, *Nonsmooth dynamic optimization of systems with varying structure*, PhD thesis, Massachusetts Institute of Technology, 2011.

[222] A. A. Zaplatin, F. K. Samigullin, V. V. Zharkov, L. N. Nikitina, and A. P. Kafeengauz, *Kinetics of the reaction of phenyl isocyanate with butanol in bipolar aprotic solvents*, Kinetika i Kataliz, 15 (1974), pp. 1382–1387.