

**Simulation, Sensitivity Analysis, and Optimization
of Bioprocesses using Dynamic Flux Balance
Analysis**

by

Jose Alberto Gomez

B.S., Tecnológico de Monterrey (2012)

M.S., Southern Methodist University (2012)

M.S.CEP, Massachusetts Institute of Technology (2014)

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

Doctor of Philosophy in Chemical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Author

Department of Chemical Engineering

December 14, 2017

Certified by

Paul I. Barton

Lammot du Pont Professor of Chemical Engineering

Thesis Supervisor

Accepted by

Patrick S. Doyle

Robert T. Haslam Professor of Chemical Engineering

Chairman, Committee for Graduate Students

Simulation, Sensitivity Analysis, and Optimization of Bioprocesses using Dynamic Flux Balance Analysis

by

Jose Alberto Gomez

Submitted to the Department of Chemical Engineering
on December 14, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

Microbial communities are a critical component of natural ecosystems and industrial bioprocesses. In natural ecosystems, these communities can present abrupt and surprising responses to perturbations, which can have important consequences. For example, climate change can influence drastically the composition of microbial communities in the oceans, which in turn affects the entirety of the food chain, and changes in diet can affect drastically the composition of the human gut microbiome, making it stronger or more vulnerable to infection by pathogens. In industrial bioprocesses, engineers work with these communities to obtain desirable products such as biofuels, pharmaceuticals, and alcoholic beverages, or to achieve relevant environmental objectives such as wastewater treatment or carbon capture. Mathematical models of microbial communities are critical for the study of natural ecosystems and for the design and control of bioprocesses. Good mathematical models of microbial communities allow scientists to predict how robust an ecosystem is, how perturbed ecosystems can be remediated, how sensitive an ecosystem is with respect to specific perturbations, and in what ways and how fast it would react to environmental changes. Good mathematical models allow engineers to design better bioprocesses and control them to produce high-quality products that meet tight specifications.

Despite the importance of microbial communities, mathematical models describing their behavior remain simplistic and only applicable to very simple and controlled bioprocesses. Therefore, the study of natural ecosystems and the design of complex bioprocesses is very challenging. As a result, the design of bioprocesses remains experiment-based, which is slow, expensive, and labor-intensive. With high-throughput experiments large datasets are generated, but without reliable mathematical models critical links between the species in the community are often missed. The design of novel bioprocesses rely on informed guesses by scientists that can only be tested experimentally. The expenses incurred by these experiments can be difficult to justify. Predictive mathematical models of microbial communities can provide insights about the possible outcomes of novel bioprocesses and guide the experimental design, resulting in cheaper and faster bioprocess development.

Most mathematical models describing microbial communities do not take into account the internal structure of the microorganisms. In recent years, new knowledge of the internal structures of these microorganisms has been generated using high-throughput DNA sequencing. Flux balance analysis (FBA) is a modeling framework that incorporates this new information into mathematical models of microbial communities. With FBA, growth and exchange flux predictions are made by solving linear programs (LPs) that are constructed based on the metabolic networks of the microorganisms. FBA can be combined with the mathematical models of dynamical biosystems, resulting in dynamic FBA (DFBA) models. DFBA models are difficult to simulate, sensitivity information is challenging to obtain, and reliable strategies to solve optimization problems with DFBA models embedded are lacking. Therefore, the use of DFBA models in science and industry remains very limited.

This thesis makes DFBA simulation more accessible to scientists and engineers with DFBA_{lab}, a fast, reliable, and efficient Matlab-based DFBA simulator. This simulator is used by more than a 100 academic users to simulate various processes such as chronic wound biofilms, gas fermentation in bubble column bioreactors, and beta-carotene production in microalgae. Also, novel combinations of microbial communities in raceway ponds have been studied. The performance of algal-yeast co-cultures and more complex communities for biolipids production has been evaluated, gaining relevant insights that will soon be tested experimentally. These combinations could enable the production of lipids-rich biomass in locations far away from power plants and other concentrated CO₂ sources by utilizing lignocellulosic waste instead.

Following reliable DFBA simulation, the mathematical theory required for sensitivity analysis of DFBA models, which happen to be nonsmooth, was developed. Methods to compute generalized derivative information for special compositions of functions, hierarchical LPs, and DFBA models were generated. Significant numerical challenges appeared during the sensitivity computation of DFBA models, some of which were resolved. Despite the challenges, sensitivity information for DFBA models was used to solve for the steady-state of a high-fidelity model of a bubble column bioreactor using nonsmooth equation-solving algorithms.

Finally, local optimization strategies for different classes of problems with DFBA models embedded were generated. The classes of problems considered include parameter estimation and optimal batch, continuous steady-state, and continuous cyclic steady-state process design. These strategies were illustrated using toy metabolic networks as well as genome-scale metabolic networks. These optimization problems demonstrate the superior performance of optimizers when reliable sensitivity information is used, as opposed to approximate information obtained from finite differences.

Future work includes the development of global optimization strategies, as well as increasing the robustness of the computation of sensitivities of DFBA models. Nevertheless, the application of DFBA models of microbial communities for the study of natural ecosystems and bioprocess design and control is closer to reality.

Thesis Supervisor: Paul I. Barton

Title: Lamot du Pont Professor of Chemical Engineering

Acknowledgments

“If I have seen further than others, it is by standing upon the shoulders of giants.”

ISAAC NEWTON

Completing a doctoral degree brings an ending to a very meaningful chapter in my life. Such an ending presents an opportunity to reflect on how much I have achieved and how far I have come in these past five years and a half, and most importantly, express my endless gratitude to those who have accompanied me closely in this journey. Earning a PhD degree can be deceiving as it often seems to be the result of the very hard work, intelligence, ingenuity, creativity, and persistence of a single person and no one else: the doctoral candidate. In my case, nothing could be further from the truth. I have been able to complete this dream with the unwavering support, patience, and kindness of many that have shared very needed words of wisdom, or have spent time with me to help me grow in different areas of my life: intellectual, professional, personal, and spiritual. These words are a heartfelt tribute to them.

First, I want to thank my research advisor, Prof. Paul I. Barton, for taking me into his group and for sharing with me his passion for mathematical modeling and optimization. During this time, he has asked me to give my best and challenged my expectations of what I thought was possible. His very high standards for the quality of work produced in the group and his care for details has allowed me to produce my best work, and his insights have given me a clear path forward when I needed guidance on my research project. In addition, he has provided me with numerous professional opportunities to meet the leaders in the field and present my work in conferences around the world. He has also opened the doors of his home several times to enjoy fun nights with present and former members of the group. Finally, he has secured the financial support needed for me to pursue my graduate degree without worries. Paul has influenced my career in a positive way and I have become a better chemical engineer in the process. For all of this I am extremely grateful.

Next I want to thank Prof. George Stephanopoulos and Prof. Chris Love, members of my thesis committee. They both participated actively in my committee meetings, asked insightful questions, and provided guidance. In particular, I want to thank Prof. Stephanopoulos for taking the time to meet a few times one on one to provide research guidance and career advice. Also, I want to thank Prof. Román for checking in every now and then to verify that things were running smoothly. I want to thank all the members of the student office. They all have been willing to listen when things were not going so great and help with administrative needs. In addition, I want to express my gratitude to Angelique for her help as the administrative assistant of Prof. Barton. I want to thank the Practice School program, and in particular Dr. Robert Hanlon, for the many lessons learned on professionalism, work ethic, and the industrial work environment during my stations in Corning and Alcon. The Practice

School program provided me with valuable industrial experience that broadened my perspective as an engineer. Also my gratitude goes to my undergraduate chemical engineering professors who motivated me to undertake this journey in the first place.

During my PhD, I have enjoyed the presence of many enthusiastic and passionate labmates that have made my time at work much better, some of which I am very fortunate enough to call friends. First, I want to mention Kai for helping me get started on my project. He was very patient and was always willing to answer questions, teach me new concepts, and share ideas on how to tackle my research project. In addition, we have shared fun sailing outings and dinners. I also want to thank Stuart and Kamil for teaching me key concepts necessary to start my research. I want to thank Garrett for listening to my many research and non-research conversations as well as for going out on social outings with me, and Peter, Michael, and Amir for sharing fun trips and outings together. During the tough times my labmates have accompanied me and provided support, keeping me on track to completing my research project while having much needed fun in the process.

When I arrived in Boston, I met my classmates which would eventually become very close friends to me. I want to thank each one of you personally. I will not list all the names for the sake of space, but each one of you has been very important to me. I also want to thank all of you who shared the Practice School experience with me. It was a very intense time, that allowed me to get to know you in special ways and become closer friends. I want to thank Siah and Rohit especially for sharing with me not only the first year classes, but for also being my labmates.

In the course of my PhD, I have been blessed to live with Justin, Harry, and Abel. You shared with me your true selves, your dreams and your fears and lent an ear when I truly needed it. You have seen me in my ups and downs and I have shared with you my joys and successes, as well as my moments of despair and of sadness. You have been my family far away from home. I am very grateful for this and I will never forget it. I want to thank Harry especially for also being classmate, labmate, Practice School group mate, travel buddy, and partying/drinking buddy. I really hope that we all remain close in the years to come.

Being far from home, I have grown close to many that have gone on a similar journey. I want to thank my Mexican friends in Boston: Juan Manuel and his friends, Paul and Miriam, Andrés, Fernando, Checo, Luly, Mariana, Lissy, Diego, Mario, Guillermo, Armida, and Ricardo. Together, we have created a warm little Mexico for us in the sometimes very chilly Boston days. I have been so lucky to share this time with all of you guys and to be able to pursue together our many dreams that have brought us to such a special place. My conversations with you have helped me remain excited about the PhD program, and have kept me thinking of ways of giving back to our common home: Mexico. I also want to thank all friends outside of MIT, such as my dancing friends and church friends, that have made of Boston my true home.

In Mexican culture, it is of paramount importance to remember where you come from and honor your roots. I have been far from home for almost six years now. In the process, I have grown apart from some of my friends at home. I want to thank Juan Manuel, Cruz, Roberto, Poncho, Rafael, Urbano, Lorenzo, Caty, Laura, and Jahaziel for making a special effort to remain close. These friends have helped me remain true

to myself, have kept me real, and have constantly reminded me where I come from. Their support has been unwavering, especially in times of need. The visits to Boston of some of you are moments I will always cherish in my heart. Your closeness and warmth disappeared the physical distance between us on numerous occasions.

También quiero agradecer a mi familia por su apoyo constante e ininterrumpido durante estos cinco años y medio. Quiero empezar por todos mis tíos y primos que me han ayudado a mantenerme emocionado respecto a mi doctorado. Quiero agradecer a mis abuelos Ramiro y Humberto, que la vida se ha llevado antes de poder terminar este logro. Sé que estarían muy orgullosos de mí y dedico este logro a honrar su memoria. Quiero agradecer a mis abuelas Evangelina y Yoyita por ser mis más entusiastas porristas durante este tiempo. A mi hermana Beatriz le quiero dar las gracias por mantenerse siempre optimista, por visitarme varias veces, por estar orgullosa de mí y por ayudarme a ser una mejor persona enseñándome a ser un mejor hermano. Quiero agradecer el apoyo incansable de mis padres José Alberto y Beatriz. No tengo palabras para describir lo importante que han sido para mí y quiero que sepan que este logro es tan suyo como mío. Sus enseñanzas, valores, y ética de trabajo que me han inculcado todo este tiempo me han permitido llegar hasta aquí. Esta tesis la dedico a toda mi familia.

Finalmente quiero agradecer a Dios por las muchas bendiciones y el mucho amor que he recibido en esta vida. He sido obstinado en mis maneras, pero Dios siempre ha encontrado la forma de encaminar mi vida de acuerdo a sus planes. En la soledad que implica la distancia de mi familia, mi cultura, y mis amigos, siempre me he sentido sostenido por el amor infinito de Dios. He encontrado apoyo en los lugares más recónditos, y centelleos de alegría y felicidad en muchos momentos inesperados. La vida ha sido muy buena conmigo. ¡Nunca olvidaré todo el apoyo y amor que he recibido de los verdaderos gigantes en mi vida que me han permitido llegar mucho más lejos de lo que jamás imaginé!

Caminante incansable: ¡detente y vuelve hacia atrás un momento!

Mira al pasado directo a los ojos,
y aprecia lo mucho que has alcanzado.

Estás en la cima de la montaña,
y todo es claro a tus pies.

Disfruta el sol y aspira el perfume de las flores.
Date un momento para sonreír y brincar con gozo.
Honra a quienes te han acompañado en tu camino,
para que tú puedas alcanzar esta hermosa cumbre.

Has llegado a tu destino caminante,
y en el destino has descubierto la semilla de un nuevo comienzo.

Despliega seguro las alas de tu alma navegante.
¡Deja a tu espíritu volar venturoso!

Ten la confianza de que siempre estarán a tu lado,
todos aquellos a quienes estás unido,
por eternos lazos de amor infinito.

José Alberto Gómez Roldán

Contents

1	Introduction	23
1.1	Optimization of DFBA models	32
1.2	Contributions and Thesis Structure	33
2	Background	35
2.1	Mathematical Preliminaries	35
2.2	Algal biofuels	42
3	DFBALab: A fast and reliable MATLAB code for Dynamic Flux Balance Analysis	47
3.1	Implementation	49
3.1.1	Lexicographic optimization	50
3.1.2	LP Feasibility Problem	53
3.1.3	Reformulation as a DAE system	55
3.2	Results and Discussion	56
3.2.1	Discussion	67
3.3	Conclusions	68
4	Modeling of an algae cultivation system for biofuels production using dynamic flux balance analysis	69
4.1	Methods	70
4.1.1	Dynamic Flux Balance Analysis	70
4.1.2	High-Rate Algal Pond Model	71

4.1.3	Raceway Open Ponds	72
4.1.4	Metabolic Models	75
4.1.5	Kinetic Parameters	77
4.1.6	Solution Equilibrium	80
4.1.7	DFBALab Hierarchy of Objectives	81
4.2	Results and Discussion	81
4.2.1	Algae monoculture without CO ₂ sparging	81
4.2.2	Algae monoculture with CO ₂ sparging	82
4.2.3	Oleaginous yeast growing on glucose	87
4.2.4	Algae/yeast coculture with cellulosic glucose feed	88
4.2.5	Algae/yeast coculture with cellulosic glucose and xylose feed and no acetate production	89
4.2.6	Algae/yeast coculture with cellulosic glucose and xylose feeds with acetate production	93
4.2.7	Economic Analysis	96
4.3	Conclusions	100
5	Multispecies Raceway Pond Modeling	103
5.1	Materials and methods	103
5.1.1	Metabolic Network Reconstructions	103
5.1.2	The Raceway Pond Model	104
5.1.3	Dynamic Flux Balance Analysis	108
5.1.4	Economic Analysis	109
5.1.5	Key differences from work in [45] and Chapter 4	109
5.2	Results	110
5.2.1	Case 1: <i>C. reinhardtii</i> and <i>E. coli</i> coculture	110
5.2.2	Case 2: <i>C. reinhardtii</i> , <i>E. coli</i> , and <i>R. glutinis</i> cultivation.	111
5.2.3	Case 3: <i>C. reinhardtii</i> and <i>R. glutinis</i> coculture	112
5.2.4	Case 4: <i>C. reinhardtii</i> and <i>S. cerevisiae</i> coculture	114
5.3	Conclusions	117

6	Sensitivities of Lexicographic Linear Programs	121
6.1	Definition of LLPs	123
6.2	Piecewise linear and piecewise affine functions	126
6.3	Extensions of directional derivatives	128
6.4	LD-derivatives of lexicographic linear programs	136
6.4.1	Computation of LD-derivatives of LLPs	138
6.4.2	Phase I LP as an extended system	147
6.5	Implementation of LD-derivatives in nonsmooth equation solving algorithms	149
6.6	Conclusions	163
7	Sensitivities of Dynamic Flux Balance Analysis Models	165
7.1	Introduction	165
7.2	Sensitivities of ODE systems with LLPs embedded	168
7.2.1	LD-derivatives of ODE systems with LLPs embedded	168
7.2.2	LD-derivatives of LLPs using the approach in Chapter 6	169
7.2.3	Alternative methods to compute LD-derivatives of LLPs	171
7.3	Efficient integration of ODE (7.6) to obtain the LD-derivatives of ODE systems with LLPs embedded	181
7.3.1	Reformulation of ODE (7.6) into a DAE system	184
7.4	Integration procedure of ODE systems corresponding to the LD-derivatives of ODE (7.1)	202
7.5	Numerical Examples	202
7.5.1	<i>E. coli</i> cultivation system	202
7.5.2	<i>E. coli</i> /yeast continuous cultivation system	206
7.6	Conclusions	206
8	Local Optimization of Dynamic Flux Balance Analysis Models	209
8.1	Toy Metabolic Network	210
8.1.1	Parameter Estimation Problem	211
8.1.2	Optimal design of a batch process	220

8.1.3	Optimal Design of a Continuous Process Operating at Steady State	223
8.1.4	Optimal Design of a Continuous Process Operating at Cyclic Steady State	231
8.1.5	Optimization of a continuous steady state process using genome-scale metabolic networks	233
8.2	Conclusions	235
9	Conclusions and Future Work	237
A	Dynamic Flux Balance Analysis using DFBAlab	243
A.1	Materials	245
A.2	Methods	246
A.2.1	Converting a GENRE in SBML format into “.mat” format	246
A.2.2	Inputs for the main.m file	247
A.2.3	Sample inputs for main.m	252
A.2.4	Inputs for the DRHS.m file	257
A.2.5	Sample inputs for the DRHS.m file	258
A.2.6	Inputs for the RHS.m file	261
A.2.7	Sample inputs for the RHS.m file	262
A.2.8	Inputs for the evts.m file	265
A.3	Notes	267
B	Extensions of Proposition 4.12 in Bonnans and Shapiro [14]	269
C	Convex and Concave Relaxations of Linear Programs and Lexicographic Linear Programs.	275
C.1	Preliminaries	275
C.1.1	McCormick’s composition theorem	276
C.2	Convex and concave relaxations for LPs	280
C.3	Convex and concave relaxations of compositions of h	284

C.4	Procedure to Calculate Convex and Concave Relaxations of $f \circ h \circ \mathbf{b}$ on P	285
C.5	Extension to Lexicographic LPs	286
C.6	Examples	289
C.6.1	Concave envelope of an LP with respect to its right-hand side	289
C.6.2	Convex and concave relaxations of factorable functions with an LP embedded	291
C.6.3	Convex and Concave Relaxations of factorable functions with a Lexicographic LP embedded	294
C.7	Conclusions	300

List of Figures

1-1	Mathematical model of a high-rate algal-bacterial pond.	24
1-2	Dynamic changes of the vaginal microbiome.	26
1-3	Typical growth curve for a bacterial population in a batch culture. . .	27
1-4	Aerobic and anaerobic growth modes for <i>E. coli</i>	28
1-5	Graphical representation of FBA.	30
1-6	Number and level of detail of published GENREs from 1999 to 2012.	31
1-7	Vision for the future of optimal design of bioprocesses.	33
2-1	Energy density comparison of several transportation fuels (indexed to gasoline = 1).	43
3-1	Concentration profiles (left) and DFBAlab penalty function (right) of Example 3.2.1.	57
3-2	DyMMM simulation results of Example 3.2.2.	61
3-3	DFBAlab simulation results of Example 3.2.2.	62
3-4	DFBAlab simulation results of Example 3.2.3.	65
3-5	Equilibrium species and pH of Example 3.2.3.	66
4-1	Photovoltaic Solar Resource of the United States.	75
4-2	Main reactions considered in the modified models iRC1080 and iND750.	78
4-3	Schematic of the raceway pond model.	82
4-4	Concentration profiles of an algae monoculture pond with no CO ₂ sparging.	83

4-5	Schematic of the algal biomass cultivation system using three raceway ponds.	84
4-6	Concentration profiles of an algae cultivation system using three raceway ponds with flue gas sparging.	85
4-7	Biomass and lipids concentrations in an algae cultivation system using three raceway ponds with flue gas sparging.	86
4-8	Algal biomass/lipids productivity and carbon balance for different sparging rates.	87
4-9	Schematic of the algae/yeast cultivation system using cellulosic glucose and three raceway ponds.	89
4-10	Concentration profiles of an algae/yeast cultivation system using three raceway ponds with cellulosic glucose.	90
4-11	Yeast, algae and lipids concentrations in a cultivation system using three raceway ponds with cellulosic glucose feed.	91
4-12	Schematic of the algal biomass cultivation system using three raceway ponds and cellulosic glucose and xylose feeds.	91
4-13	Concentration profiles of an algae/yeast cultivation system using three raceway ponds with cellulosic glucose and xylose feeds.	92
4-14	Yeast, algae and lipids concentrations in an algae/yeast cultivation system using three raceway ponds with cellulosic glucose and xylose feeds.	93
4-15	Schematic of the algal biomass cultivation system using three raceway ponds and cellulosic glucose and xylose feeds with acetate production.	94
4-16	Concentration profiles of an algae/yeast cultivation system using three raceway ponds with cellulosic glucose and xylose feeds with acetate production.	95
5-1	Schematic of the interactions amongst <i>E. coli</i> , <i>C. reinhardtii</i> , <i>S. cerevisiae</i> , and <i>R. glutinis</i>	105
5-2	pH factors for different microorganisms.	107

5-3	Three pond system for microbial cultivation.	110
5-4	Concentrations of substrates, nutrients, and products in the three-pond system for Case 1.	112
5-5	Concentrations of substrates, nutrients, and products in the three-pond system for Case 2.	113
5-6	Concentrations of substrates, nutrients, and products in the three-pond system for Case 3.	114
5-7	Evolution of concentrations after the appearance of 1 mg/L of <i>E. coli</i> in the first pond with no pH control.	115
5-8	Evolution of concentrations after appearance of 1 mg/L of <i>E. coli</i> in the first pond with modified feeds to control pH.	116
5-9	Concentrations of substrates, nutrients, and products in the three-pond system for Case 4.	117
5-10	Concentration of biomass, ethanol, and lipids in each pond for each case.	118
6-1	Graphical explanation of LD-derivatives for Example 6.4.1.	144
6-2	Surface plots of \mathbf{h} with respect to \mathbf{b} in Example 6.4.1.	146
7-1	Sensitivities plots for Example 7.3.2.	197
7-2	DFBA simulation and sensitivities for a batch process growing <i>E. coli</i> on glucose and xylose.	205
7-3	DFBA simulation and sensitivities for a continuous process involving <i>E. coli</i> and yeast.	207
8-1	Simulated and experimental data for biomass and lipids for a batch experiment using the toy metabolic network.	216
8-2	Simulated and experimental data for substrates and products for a batch experiment using the toy metabolic network.	217
8-3	Simulated and experimental data for biomass and lipids for a batch experiment using the toy metabolic network.	218

8-4	Simulated and experimental data for substrates and products for a batch experiment using the toy metabolic network.	219
8-5	Biomass and lipids concentrations for optimal batch parameters.	222
8-6	Substrate and product concentrations for optimal batch parameters.	223
8-7	Biomass and lipids concentrations for optimal batch parameters.	225
8-8	Substrate and product concentrations for optimal batch parameters.	226
9-1	Vision for the future of bioprocess design.	242
C-1	Convex and concave envelopes for function (C.17).	290
C-2	Convex and concave relaxations of (C.18).	292
C-3	Convex and concave relaxations of (C.19).	293
C-4	Upper and lower bounds of (C.19).	294
C-5	Convex and concave relaxations of (C.19) on smaller sections.	295
C-6	Plots of $\mathbf{h} \circ \mathbf{b}$ and $f \circ \mathbf{h} \circ \mathbf{b}$	297
C-7	Plots of $\mathbf{h} \circ \mathbf{b}$ and $f \circ \mathbf{h} \circ \mathbf{b}$ with convex relaxations.	298
C-8	Plots of $\mathbf{h} \circ \mathbf{b}$ and $f \circ \mathbf{h} \circ \mathbf{b}$ with concave relaxations.	299

List of Tables

3.1	Initial concentrations and parameters of Example 3.2.2.	59
3.2	Priority list order for the lexicographic linear programs in Example 3.2.2.	59
3.3	Initial concentrations and parameters of Example 3.2.3.	65
3.4	Running times of Example 3.2.4 with increasing number of models.	66
4.1	Summary of uptake kinetic parameters for algae and yeast.	79
4.2	Constants for pH dependent uptakes of algae.	79
4.3	Hierarchy of objectives for simulation with DFBAlab.	81
4.4	Carbon balance of an algal monoculture with flue gas sparging	86
4.5	Inputs and outputs for yeast monocultures and yeast/algal cocultures with constant light.	88
4.6	Carbon balance of coculture with pure glucose feed	89
4.7	Carbon balance of coculture with glucose/xylose feed.	93
4.8	Carbon balance of a coculture with glucose/xylose feed and acetate production.	95
4.9	Economic analysis for biodiesel production using CO ₂ sparging.	99
4.10	Economic analysis for biodiesel production with pure glucose and a glucose/xylose mix.	101
4.11	Economic analysis comparison between coculture systems using a glu- cose/xylose mix.	102
5.1	Summary of uptake kinetic parameters for all microorganisms	106
5.2	Constants for pH dependent uptakes for different microorganisms	107
5.3	Hierarchy of objectives used in DFBAlab.	108

5.4	Feeds for the different pond distributions	111
5.5	Carbon balances for the different cases	115
5.6	Lipids fraction of biomass for each microorganism at each pond for all cases.	116
5.7	Economic analysis for biodiesel production.	119
6.1	Hierarchy of objectives for bubble column bioreactor	151
6.2	Number of iterations and 2-norm for Example 6.5.1 with first start point. 153	
6.3	Number of iterations and 2-norm for Example 6.5.1 with second start point.	154
6.4	Data for Firm 1 in Example 6.5.2	155
6.5	Data for Firm 2 in Example 6.5.2	155
6.6	Prices of Chemicals	156
6.7	Running times for the optimization problem (6.21).	158
6.8	Located solutions and running times for a CSTR non-smooth equation solving problem.	161
6.9	Sequence of iterates for the NFD (1) and QSNM (2) when starting at (2,0,0,0).	162
6.10	Sequence of iterates for the QSNM with no <i>E. Coli</i> feed.	162
6.11	Sequence of iterates for the LPNM with no <i>E. Coli</i> feed.	163
6.12	Located solutions and running times for a CSTR non-smooth equation solving problem.	164
6.13	Sequence of iterates for LPNM with no <i>E. Coli</i> feed.	164
8.1	Stoichiometry Matrix for Toy Metabolic Network.	212
8.2	Technology matrix for FBA problem in standard form.	213
8.3	Hierarchy of objectives for the toy metabolic network in 8.2.	214
8.4	Simulation data for Toy Metabolic Network.	214
8.5	Experimental data for Toy Metabolic Network.	215
8.6	Comparison of optimization results with initial and base points.	218

8.7	Comparison of optimization results with initial and base points when weights are used.	220
8.8	Summary of results for the optimal design of a batch process using P .	221
8.9	Summary of results for the optimal design of a batch process using \hat{P} .	224
8.10	Comparison of the optimal result with start point for continuous steady state optimization using first optimization strategy.	229
8.11	Comparison of optimization results with initial point for continuous steady state optimization using second optimization strategy.	230
8.12	Summary of results for the optimal design of a cyclic steady state system.	233
8.13	Summary of results for the optimal design of a steady state system producing ethanol with <i>E. coli</i> and yeast.	235

Chapter 1

Introduction

Due to their widespread application in industrial bioprocesses and their occurrence in natural ecosystems, microbial communities are a relevant subject of study. They can be found in very diverse industrial and natural settings such as wastewater treatment [18], pharmaceuticals from recombinant DNA technology [67], the human gut microbiome [81], the ocean ecosystem [39], among many other examples. Mathematical modeling of these systems is interesting: it allows for the control and optimal design of industrial bioprocesses, and predicts the sensitivity of the microbial community to changes in the environment in natural ecosystems. For example in [18], the authors use a mathematical model of a raceway pond to study the influence of process parameters, such as temperature and residence time, on algal yield. The mathematical model used can be seen in Figure 1-1. In [39] the authors use mathematical models to predict which class of photoautotrophs dominate different sections of the ocean. Mathematical models enable better understanding of very complex systems and provide answers to interesting questions such as the following:

1. Which are the most important parameters in the system?
2. How does the system respond to changes?
3. How stable is a steady state?
4. How to make the biosystem work better?

5. How to combine species in novel systems designed for human purposes?

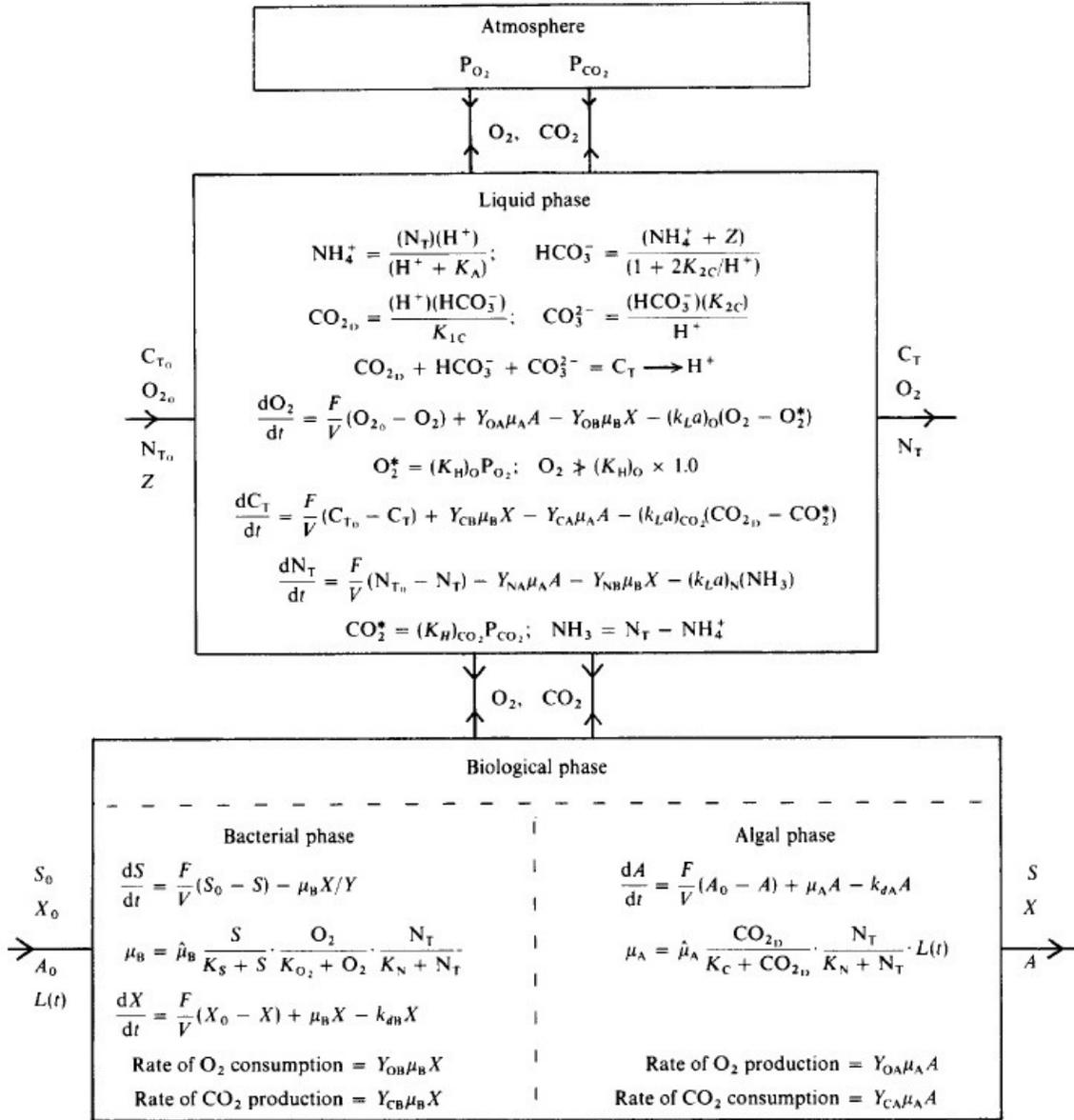


Figure 1-1: Mathematical model of a high-rate algal-bacterial pond. Reproduced from [18].

Microbial communities are difficult to model because they are complex, dynamic, and involve many symbiotic and competitive relationships that may not be obvious at first glance. An example of how much microbial communities can change with time is shown in the vaginal microbiome illustrated in Figure 1-2. Mathematical models of

microbial communities require information on growth rates and exchange flux rates of the different microorganisms involved. Despite the importance of bioprocesses, mathematical models used in industry to describe the growth and exchange fluxes rates of microorganisms remain rather simplistic. Most expressions describing growth rates rely on unstructured models. These models are called unstructured because they do not consider any structural information concerning the microorganisms, such as their metabolic network or cell compartments. One example of an unstructured model of growth is the widely-used Monod equation. Jacques Monod introduced the Monod equation to model bacterial growth in the exponential phase under a limiting substrate [94]:

$$\mu(S) = \mu_{\max} \frac{S}{K_m + S}, \quad (1.1)$$

where S refers to the limiting substrate concentration, μ_{\max} is the maximum growth rate, and K_m is the half-velocity constant. The constants of these equations can be obtained from correlating Equation (1.1) with experimental data. Other expressions that attempt to describe the growth rate of microorganisms include the Contois, Tessier, Moser, Blackman equations [122] or the Droop model [30, 88]. The variety of unstructured models provides flexibility to model different growth conditions.

Growth of microorganisms in a batch culture usually present the following phases [122] (see Figure 1-3):

1. Lag phase: this phase corresponds to a period of adaptation where cells synthesize new enzymes or repress current enzymes to better use the resources in the cultivation medium.
2. Exponential growth phase: once adapted, cells can multiply rapidly. In this phase, no substrate is limiting and cells have a constant doubling time. This is a period of balanced growth (cell mass composition is constant).
3. Deceleration phase: in this phase growth slows down due to the depletion of an essential nutrient or the accumulation of toxic by-products. This is a period

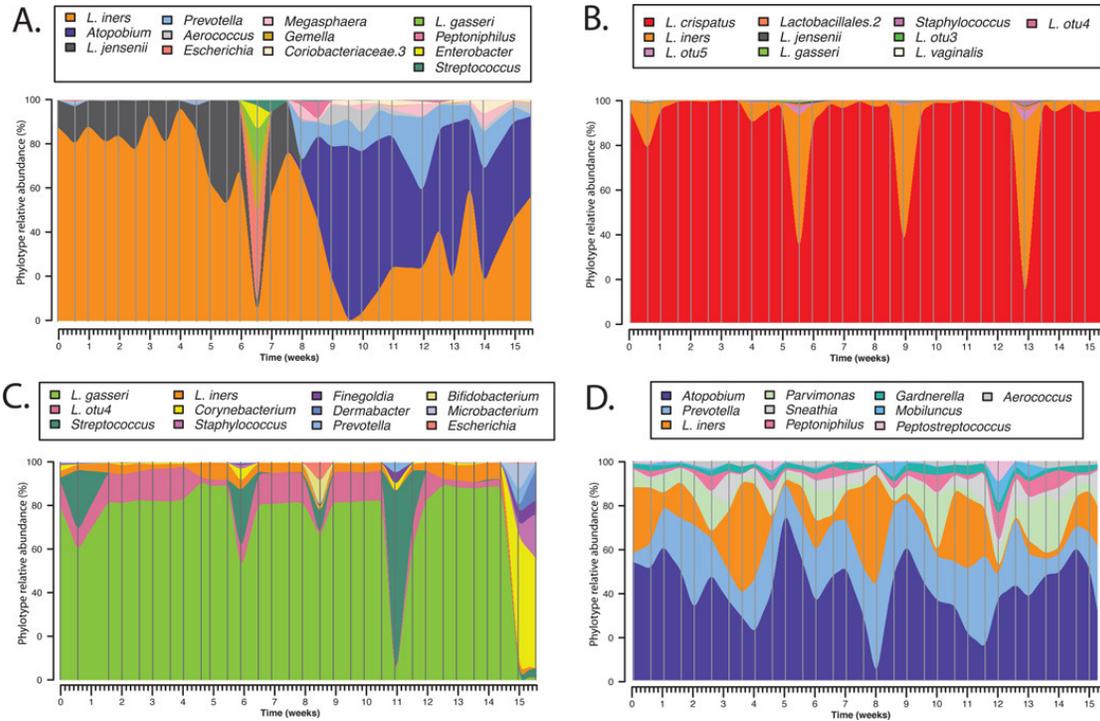


Figure 1-2: Dynamic changes of the vaginal microbiome for four subjects. Very drastic changes can be observed in all subjects. Mathematical models can be a very useful tool to determine when these changes may take place, predict the new microbiome composition, or help design drugs that would promote a specific microbiome composition. Reproduced from [85].

of unbalanced growth where cells restructure their composition to increase the prospects of cellular survival.

4. Stationary phase: this is a phase of net growth zero.
5. Death phase: this phase corresponds to all remaining cells dying due to lack of essential nutrients or buildup of toxic chemicals in the medium.

Most growth models, such as unstructured models, assume balanced growth conditions, which occur at steady-state continuous cultures or the exponential phase of batch cultures. Unstructured models can be modified to model other phases. For example, a time delay can be added to model the lag phase. Unstructured models cannot describe transient conditions [122]. Attempts to model multiple growth modes simultaneously have been made in [99, 38, 151], but these expressions grow rapidly in complexity and require a priori knowledge from the modeler of the different metabolic

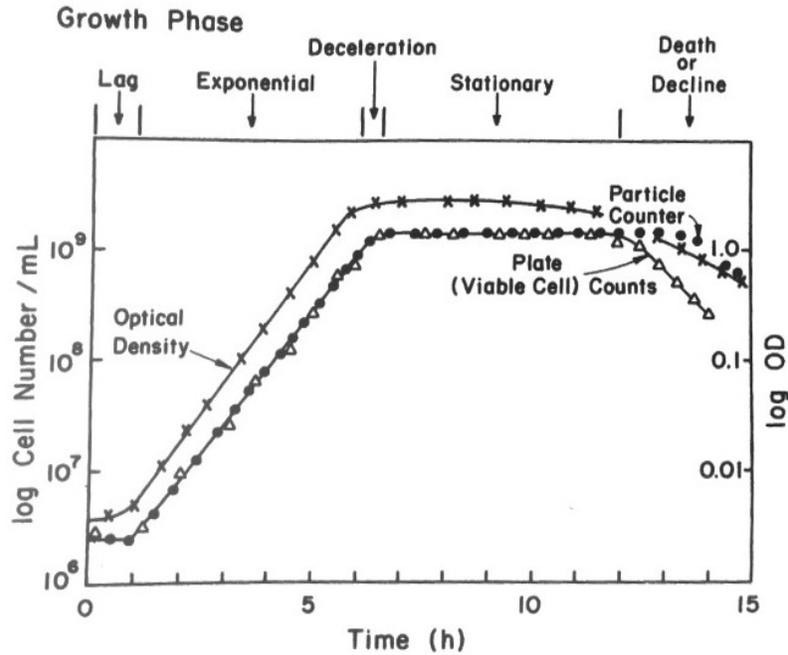


Figure 1-3: Typical growth curve for a bacterial population in a batch culture. Reproduced from [122].

states the microorganisms in the system can encounter. In particular, notice the complexity of Equations (1) to (19) in [99]. These equations were derived for the specific system described in the paper and are applicable to a system that considers three substrates and three enzymes. Any minor changes in the system, such as the interactions of two microorganisms through the exchange of a critical nutrient, would result in a different set of equations. Therefore, unstructured models cannot be used to predict the performance of novel process setups. In particular, bioprocesses where microorganisms present symbiotic or competitive relationships, grow under multiple nutrient limitations, or attain cyclic steady-states, are challenging to model because microorganisms switch between different growth modes over time. Even if all the constants for all possible growth modes were determined, it is not clear when the microorganisms switch from one growth mode described by one set of constants, to the next one described by a different set of constants (see Figure 1-4).

Therefore, given an extracellular environment, a method that selects the growth modes describing each microorganism in a culture from all possible modes is necessary.

Such a modeling framework can be provided by structured models that consider the metabolic networks of the different microorganisms. Flux balance analysis (FBA) [137, 98] does exactly this.

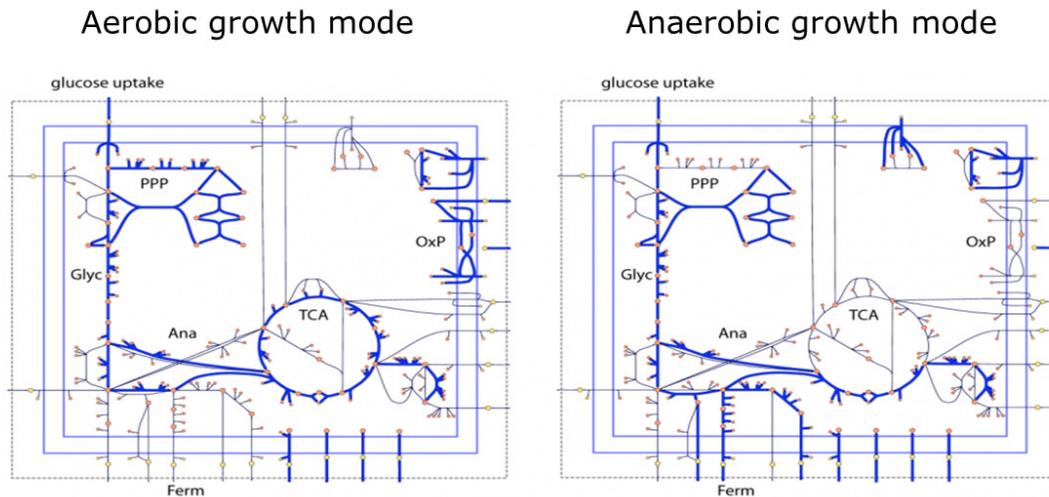


Figure 1-4: Metabolic network of *E. coli* under aerobic (left) and anaerobic (right) growth modes. The active pathways, those carrying some flux, are shown on bold blue. Under unstructured models, both growth modes are modeled using different constant values. If *E. coli* switches between the two growth modes, transition rules need to be determined. Figure reproduced from [97].

FBA is a constraint-based modeling framework that uses the information in genome-scale metabolic network reconstructions (GENREs) to predict growth and exchange fluxes rates of microorganisms. Thermodynamics impose more constraints in the form of irreversible reactions. Some other constraints can be imposed by the extracellular environment. For instance, for a microorganism that consumes O_2 , how much O_2 is available in the extracellular environment will provide an upper bound on how much can be consumed. Given this set of constraints, the system is underdetermined. However, points that maximize certain objectives can be identified [98]. Of particular interest are the points that maximize growth rate as they tend to have good agreement with experimental data. The resulting formulation can be described by a linear

program (LP):

$$\begin{aligned} \max_{\mathbf{v}} \quad & \mathbf{c}^T \mathbf{v} \\ \text{s.t.} \quad & \mathbf{S} \mathbf{v} = \mathbf{0}, \\ & \mathbf{v}^{LB} \leq \mathbf{v} \leq \mathbf{v}^{UB}, \end{aligned} \tag{1.2}$$

where \mathbf{c} is the cost function (usually maximize growth), \mathbf{v} is the flux vector, \mathbf{S} is the stoichiometry matrix that represents a GENRE, and $\mathbf{v}^{LB}, \mathbf{v}^{UB}$ are lower bounds and upper bounds, respectively, on the fluxes given by thermodynamics or by the extracellular environment. Figure 1-5 illustrates FBA graphically.

The use of LP (1.2) requires a metabolic network. Fortunately, with the advent of high-throughput DNA sequencing, more metabolic networks are now becoming available. Figure 1-6 shows how the number of models and the level of detail considered in these models has expanded considerably since 1999. A good source of published GENREs can be found at the Systems Biology Research Group in University of California San Diego [126].

A bioprocess mode described by an ordinary differential equation (ODE) system or a differential-algebraic equation (DAE) system, such as the raceway pond model described in Figure 1-1, can be combined with FBA models. This results in a dynamic FBA (DFBA) model [137, 87]. In a DFBA model, growth and exchange fluxes rates are given by the solution of the FBA model. This modeling framework is based on the assumption that intracellular dynamics equilibrate much faster than extracellular ones, and therefore, the cell is in quasi steady-state [125]. Therefore, the use of FBA provides a good approximation.

With DFBA, complex bioprocesses where microorganisms experience different growth modes, encounter multiple substrate limitations, or experience symbiotic and/or competitive relationships, can be modeled more reliably compared to unstructured models. Despite the power of this modeling framework, it is rarely used. A DFBA model results in an ODE or DAE system with LPs embedded. These systems are challenging to simulate and optimize. Methods relying on collocation, such

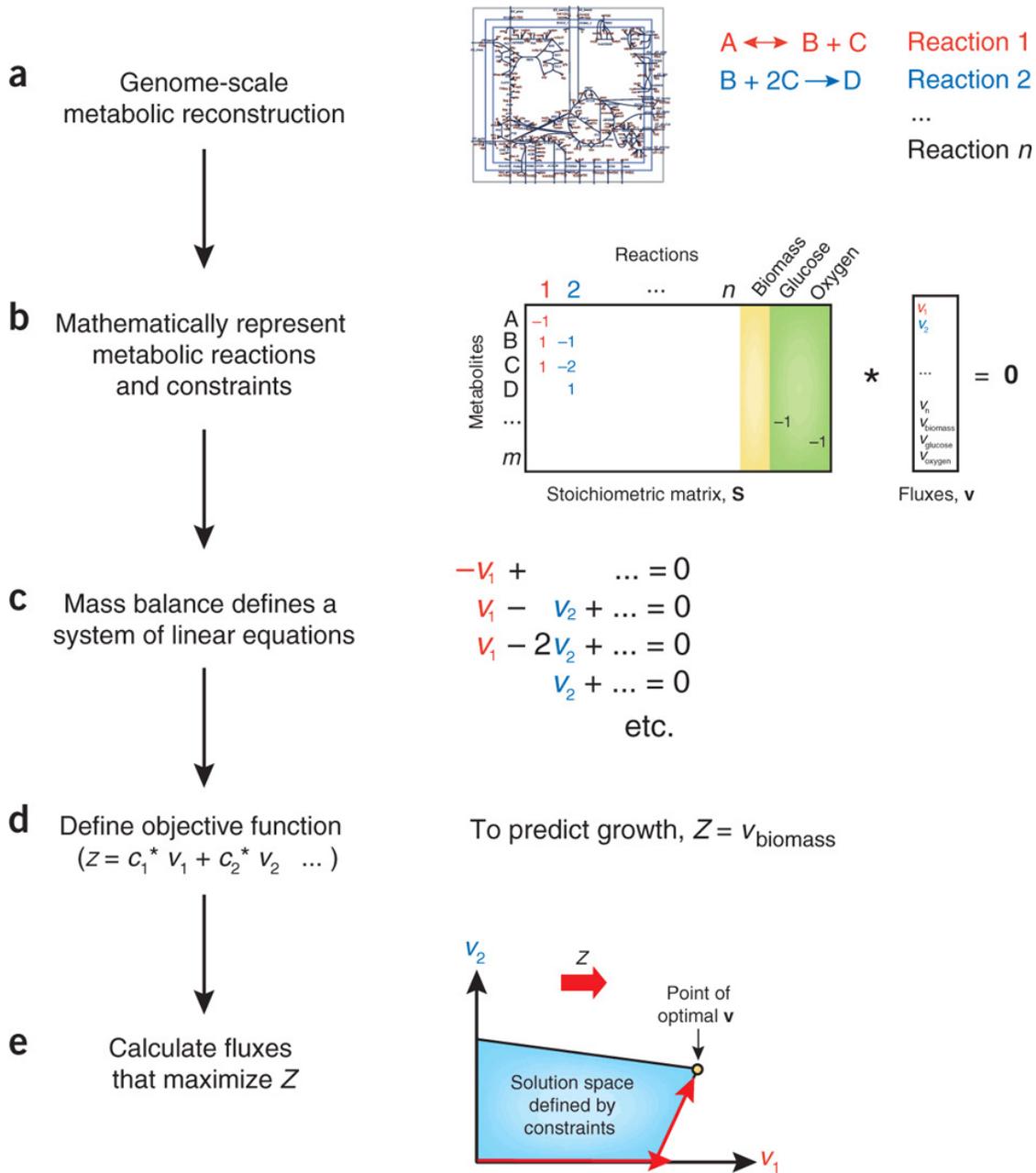


Figure 1-5: Graphical representation of FBA. Reproduced from [98].

as the one in [104], can be inaccurate because DFBA systems can be stiff; therefore, how to best discretize the time horizon is not obvious beforehand. Another approach is reformulating the LP as its Karush-Kuhn-Tucker (KKT) conditions, resulting in a DAE system [69]. However, the nonuniqueness of the LP causes this DAE system to have index greater than 1. Finally, there is the approach of using a variable time-

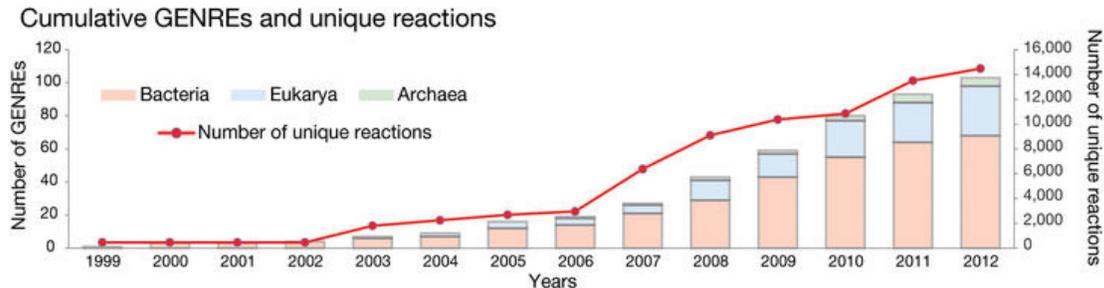


Figure 1-6: Number and level of detail of published GENREs from 1999 to 2012. Reproduced from [93].

stepping method to integrate the DFBA system and solving the LP at each time step. This method is not well-defined as the LP does not necessarily have a unique solution. This is illustrated by the following Example.

Example 1.0.1. Consider the following ODE system:

$$\begin{aligned} \dot{y}(t, p) &= v_1(y(t, p))v_2(y(t, p)) - v_2(y(t, p)), \forall t \in (t_0, t_f] \\ y(t_0, p) &= p, p \in [0, 1], \\ \mathbf{v}(z) &\in \arg \min -v_1, \text{ s.t. } v_1 \leq z, v_1 + v_2 \leq 1, \mathbf{v} \geq \mathbf{0}. \end{aligned}$$

Let $p(t_0) = 0.5$. Then, $v_1(y(t_0, p)) = 0.5$ and $v_2(y(t_0, p)) \in [0, 0.5]$. This implies that $\dot{y}(t_0, p) \in [-0.25, 0]$. This ODE system with an LP embedded is not well-defined as the right-hand side of this ODE system is set-valued.

It is important to notice that DFBA models are multi-scale. Whereas FBA considers length and time scales associated with individual cells, the process model described by the ODE, DAE, or PDE system considers length and time scales corresponding to the reactor. The time scales associated with dynamic changes in the cellular level are much faster than those occurring in the reactor. The pseudo steady-state assumption is an approximation of the very fast cellular time scales that allows to model the individual cells as LPs at all times, resulting in a nonsmooth dynamic system.

In addition, all of the methods previously described fail when the embedded LP becomes infeasible. Infeasible LPs in the context of FBA mean that there are not

enough substrates and nutrients to support growth in the medium, and therefore, the microorganisms for which the LP becomes infeasible will start dying. However, the LP becoming infeasible can cause the integrator to fail prematurely unless an extension of the feasible set is used as described in Chapter 3 of this thesis.

1.1 Optimization of DFBA models

A broad class of optimization problems for DFBA models can be defined as:

$$\begin{aligned} \min_{\mathbf{p}} J(\mathbf{p}) &\equiv \varphi(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} l(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt & (1.3) \\ \text{s.t. } \mathbf{g}(\mathbf{p}) &\equiv \mathbf{r}(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} \mathbf{s}(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt \leq \mathbf{0}, \\ \mathbf{p} &\in F \subset D_p \subset \mathbb{R}^{n_p}, \end{aligned}$$

where $D_p \subset \mathbb{R}^{n_p}$ is an open set, and F is the set of feasible parameter values, which are those that lead to feasible trajectories for the DFBA model over the entire time horizon and satisfy physical bounds. Notice that equality constraints can be modeled as a pair of inequality constraints. The functions $\mathbf{r}(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p})$ and $\mathbf{s}(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p})$ can enforce path constraints. In the case of DFBA systems, J and \mathbf{g} are nonsmooth functions. This means that the classical derivative does not exist for all points in the domain of these functions. Therefore, generalized derivatives and nonsmooth optimization techniques are needed.

The tools, algorithms, and mathematical developments developed in this thesis will bring us closer to model-based optimal bioprocess discovery. Using mathematical models, the development of new bioprocesses can be speeded up because computational experiments are faster and less-costly than bench-scale experiments. This model-based approach is illustrated in Figure 1-7.

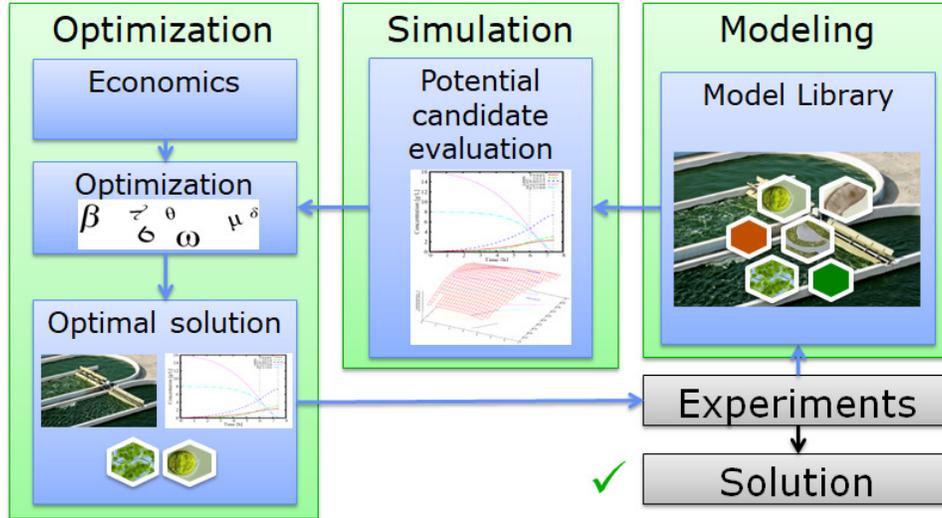


Figure 1-7: Vision for the future of optimal design of bioprocesses. A model library containing different process models and metabolic networks of different microorganisms can be created. This library allows testing different combinations of process models and microorganisms. These new setups can be simulated to predict the different outcomes. Mathematical optimization can be used to modify the parameters to improve objectives such as cost reduction, biomass accumulation, or production of specialty chemicals. The optimized designs can be tested on bench-scale. If there is good agreement between the computational and bench-scale experiments, a new optimized design has been found. Otherwise, knowledge is gathered from the bench-scale experiments to refine the models and a new optimization loop takes place. In this way, the model drives the experiments. Since mathematical optimization is faster than bench-scale optimization, an optimal design can be found in a shorter time frame.

1.2 Contributions and Thesis Structure

The main contributions of this thesis are the following:

1. The development of an efficient, reliable, and user-friendly simulator for DFBA models in Matlab.
2. The development of DFBA models of raceway ponds for biomass cultivation.
3. The mathematical derivation of sensitivities for DFBA models.
4. The development of optimization strategies for DFBA models.

The thesis is organized in the following way. In Chapter 2 the mathematical

background regarding nonsmooth analysis is introduced. Here, the concepts of lexicographic differentiation [96] and LD-derivatives [72, 74] are introduced.

Chapter 3 talks about DFBA_{lab}, a user-friendly, efficient, and reliable DFBA simulator in Matlab [44]. This chapter talks about the mathematical theory behind this simulator and illustrates its performance in different case studies.

Chapter 4 presents a DFBA model for a raceway pond used for algae cultivation [45]. This mathematical model results from the combination of the high-rate algal-bacterial pond [18, 144] and DFBA theory. Different cultivation strategies are explored including algae/yeast cocultures growing on cellulosic sugars.

Chapter 5 builds on the work of Chapter 4 to explore multispecies cultivation in raceway ponds. In addition, it adds layers of complexity to the model in [45] by making lipids accumulation in yeasts variable.

Chapter 6 develops the sensitivity theory for lexicographic LPs. In particular, it generalizes the chain rule for situations where the outer function of a composition is defined on a closed set. In addition, it generalizes this chain rule for LD-derivatives. Finally, it obtains the LD-derivatives for lexicographic LPs.

Chapter 7 applies the theory in Chapter 6 and in [72, 73] to obtain the sensitivity information for DFBA systems. The sensitivities of a DFBA model containing a genome-scale metabolic network is used to illustrate the use of this theory.

Chapter 8 describes different optimization strategies for batch, fed-batch, and continuous bioprocesses described by DFBA models. It uses the sensitivities described in Chapter 7 and nonsmooth optimization solvers as well as IPOPT to perform local optimization of DFBA models.

Finally, Chapter 9 describes the remaining challenges and the future work for the simulation and optimization of DFBA models.

Chapter 2

Background

2.1 Mathematical Preliminaries

Let all norms be the Euclidean norm. Boldface symbols represent vector and matrix-valued quantities. Let V be a subset of a metric space, then $\text{int}(V)$ and $\text{bnd}(V)$ denote the interior and the boundary of V , respectively. Let $L(\mathbb{R}^n; \mathbb{R}^m)$ be the space of linear maps from \mathbb{R}^n to \mathbb{R}^m ; each element of $L(\mathbb{R}^n; \mathbb{R}^m)$ can be identified with an $m \times n$ matrix. For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, let $\mathcal{R}(\mathbf{A}) \subset \mathbb{R}^m$ be the column space of \mathbf{A} . The i th column vector of a matrix \mathbf{M} is denoted by \mathbf{m}_i . Denote by $GL(n, \mathbb{R})$ the set of all invertible $n \times n$ matrices. Let $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ be the extended real number system. Let \mathbb{R}_+ be the nonnegative part of the real line and \mathbb{R}_- the nonpositive part of the real line. Let $\mathbf{0}$ be a vector with all components equal to zero, $\mathbf{1}$ be a vector with all components equal to 1 and \mathbf{e}_i be a vector with all components equal to zero except to the i th component which is equal to one. Let \mathbf{I}_m be the identity matrix with m rows. Consider two vectors $\mathbf{x}^1, \mathbf{x}^2 \in \mathbb{R}^m$; $\mathbf{x}^1 > \mathbf{x}^2$ if for all $i \in \{1, \dots, m\}$, $x_i^1 > x_i^2$ and $\mathbf{x}^1 \geq \mathbf{x}^2$ if for all $i \in \{1, \dots, m\}$, $x_i^1 \geq x_i^2$. Consider a set J with a finite number of elements. $\text{card}(J)$ refers to the cardinality of this set. The convex hull of a set X will be denoted as $\text{conv}(X)$. Let a function f be C^k if it is k times continuously differentiable, and PC^k if it is piecewise differentiable k times in the sense of [119].

Definition 2.1.1. [27] Let $X \in \mathbb{R}^m$ be open and let $\mathbf{x} \in X$. A function $f : X \rightarrow \mathbb{R}$

is said to be *Lipschitz near* \mathbf{x} if there exists a neighborhood $N_\delta(\mathbf{x})$ of \mathbf{x} and $K > 0$ such that

$$|f(\mathbf{y}) - f(\mathbf{x})| \leq K\|\mathbf{y} - \mathbf{x}\|,$$

for all $\mathbf{y} \in N_\delta(\mathbf{x})$. A function is said to be *locally Lipschitz* on X if it is Lipschitz near $\hat{\mathbf{x}}$ for any $\hat{\mathbf{x}} \in X$ [119].

Vector-valued functions are locally Lipschitz continuous if all their components are locally Lipschitz continuous.

Definition 2.1.2. Let $X \subset \mathbb{R}^n$ be an open set and $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. The (one-sided) *directional derivative* of \mathbf{f} at $\mathbf{x} \in X$ in the direction $\mathbf{d} \in \mathbb{R}^n$ is given by the following limit if it exists:

$$\mathbf{f}'(\mathbf{x}; \mathbf{d}) \equiv \lim_{\tau \rightarrow 0^+} \frac{\mathbf{f}(\mathbf{x} + \tau\mathbf{d}) - \mathbf{f}(\mathbf{x})}{\tau}.$$

If at \mathbf{x} , the limit exists in \mathbb{R}^n for all directions $\mathbf{d} \in \mathbb{R}^n$, then, \mathbf{f} is said to be *directionally differentiable* at \mathbf{x} .

For the remaining definitions, assume $X \subset \mathbb{R}^n$ is an open set and $\mathbf{f} : X \rightarrow \mathbb{R}^m$ is locally Lipschitz continuous. Next the definition of the classical derivative is introduced.

Definition 2.1.3. [27] \mathbf{f} is (Gâteaux) differentiable at $\mathbf{x} \in X$ if there exists a unique *derivative* $\mathbf{Jf}(\mathbf{x}) \in \mathbb{R}^{m \times n}$ for which

$$\mathbf{Jf}(\mathbf{x})\mathbf{d} = \lim_{\tau \rightarrow 0^+} \frac{\mathbf{f}(\mathbf{x} + \tau\mathbf{d}) - \mathbf{f}(\mathbf{x})}{\tau}, \quad \forall \mathbf{d} \in \mathbb{R}^n.$$

This derivative corresponds to the *Jacobian matrix* of \mathbf{f} at \mathbf{x} . In this case (locally Lipschitz continuous), the Gâteaux and Fréchet derivatives are equal.

The mathematical work in this thesis requires the theory of nonsmooth functions. Nonsmooth functions are those for which the classical derivative does not exist everywhere. Therefore, we now introduce some generalizations of the derivative for

nonsmooth functions. For locally Lipschitz continuous functions, Rademacher's Theorem guarantees the differentiability of \mathbf{f} at each point in $X \setminus Z_{\mathbf{f}}$ where $Z_{\mathbf{f}} \subset X$ is some set of measure zero [27].

Definition 2.1.4. [27] The Bouligand (*B-*)*subdifferential* is defined as

$$\partial_B \mathbf{f}(\mathbf{x}) \equiv \{\mathbf{H} \in \mathbb{R}^{m \times n} : \mathbf{H} = \lim_{i \rightarrow \infty} \mathbf{J}\mathbf{f}(\mathbf{x}_{(i)}), \mathbf{x} = \lim_{i \rightarrow \infty} \mathbf{x}_{(i)}, \mathbf{x}_{(i)} \in X \setminus Z_{\mathbf{f}}, \forall i \in \mathbb{N}\}.$$

Definition 2.1.5. [27] The Clarke (*generalized*) *Jacobian* of \mathbf{f} at $\mathbf{x} \in X$ is

$$\partial \mathbf{f}(\mathbf{x}) \equiv \text{conv}(\partial_B \mathbf{f}(\mathbf{x})).$$

When the function is continuously differentiable, the generalized Jacobian results in a singleton corresponding to the classical derivative.

Example 2.1.1. Consider $f(x) : x \mapsto |x|$. The derivative of f at $x = 0$ is not defined in the classical sense. However, the B-subdifferential and the generalized Jacobian are defined: $\partial_B f(0) = \{-1, 1\}$ and $\partial f(0) = [-1, 1]$. Notice that for all $x \neq 0$, $\{f'(x)\} = \partial_B f(x) = \partial f(x)$.

Nonsmooth optimization [89] and equation-solving algorithms [103, 33] have been designed to take elements of the generalized Jacobian as inputs. However, using the generalized Jacobian presents a difficulty: it does not satisfy a sharp chain rule. In general, for $\mathbf{h} : \mathbb{R}^m \rightarrow \mathbb{R}^l$ and for $\mathbf{x} \in X$

$$\partial[\mathbf{h} \circ \mathbf{f}](\mathbf{x}) \subset \text{conv}(\{\mathbf{H}\mathbf{F} : \mathbf{H} \in \partial \mathbf{h}(\mathbf{f}(\mathbf{x})), \mathbf{F} \in \partial \mathbf{f}(\mathbf{x})\}). \quad [27]$$

Therefore, applying the chain rule does not allow finding an element of the generalized derivative of a composition of functions. Other calculus rules such as the sum rule fail by the same reason. This can be seen in the following example.

Example 2.1.2. Consider $f(x) = g(x) + h(x)$ where $g(x) : x \mapsto \min(0, x)$ and $h(x) : x \mapsto \max(0, x)$. It is clear that $f(x) = x$ and therefore $\{f'(x)\} = \partial f(x) = \{1\}$

for all $x \in \mathbb{R}$. Now consider $x = 0$. Then, $\partial g(0) = \partial h(0) = [0, 1]$. Notice that $\partial f(0)$ is a strict subset of $\partial g(0) + \partial h(0) = [0, 2]$.

Example 2.1.3. Consider $g(x) : x \mapsto \max(0, x)$, $h(x) : x \mapsto \min(0, x)$ and $f = [h \circ g]$. It is clear that $f(x) = 0$ for all $x \in \mathbb{R}$ and therefore $\{f'(x)\} = \partial f(x) = \{0\}$. Consider $x = 0$. Then $g(0) = 0$, $\partial g(0) = [0, 1]$ and $\partial h(0) = [0, 1]$. Applying the chain rule results in,

$$\partial h(g(0))\partial g(0) = [0, 1], \tag{2.1}$$

which is an overestimation of $\partial f(0) = \{0\}$.

In addition, elements of the generalized derivative cannot be estimated using finite differences in the coordinate directions. This is shown by the following example.

Example 2.1.4. Consider $f(\mathbf{x}) = 0.5|x_1 + x_2| + 0.5|x_1 - x_2|$. This function is non-smooth at all points $x_1 = x_2$ and $x_1 = -x_2$. Consider $\mathbf{x} = \mathbf{0}$. Then, $\partial_B f(\mathbf{0}) = \left\{ \begin{bmatrix} 1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \end{bmatrix} \right\}$. If we take the directional derivatives in the coordinate directions, we get:

$$\begin{bmatrix} f'(\mathbf{0}; \mathbf{e}_1) & f'(\mathbf{0}; \mathbf{e}_2) \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} \notin \text{conv}(\partial_B f(\mathbf{0})).$$

In addition, the Clarke Jacobian may be a strict subset of the Cartesian product of the componentwise Clarke gradients.

Example 2.1.5. Consider $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2 : (x_1, x_2) \mapsto (x_1 + |x_2|, x_1 - |x_2|)$. Let $\mathbf{x} = \mathbf{0}$. Then,

$$\partial \mathbf{f}(\mathbf{0}) = \left\{ \begin{bmatrix} 1 & 2\lambda - 1 \\ 1 & 1 - 2\lambda \end{bmatrix} \forall \lambda \in [0, 1] \right\},$$

and

$$\partial f_1(\mathbf{0}) \times \partial f_2(\mathbf{0}) = \left\{ \begin{bmatrix} 1 & 2\lambda_1 - 1 \\ 1 & 2\lambda_2 - 1 \end{bmatrix} \forall \lambda_1, \lambda_2 \in [0, 1] \right\} \supset \partial \mathbf{f}(\mathbf{0}).$$

These properties make it computationally difficult to obtain elements of the generalized derivative. Therefore, solving nonsmooth equation and optimization problems is considered difficult. In general, people in the field have tried different strategies to relax the nonsmoothness resulting in complex models that are difficult to relate to physical quantities and additional parameters that explode in number.

Fortunately Nesterov [96] and Khan and coworkers [72, 74] have introduced the concept of lexicographic derivatives and lexicographic directional derivatives, respectively. These generalizations of the derivative and the directional derivative present very amenable properties. We next introduce them in the following definitions.

Definition 2.1.6. [96] Let $X \subset \mathbb{R}^n$ be open and $\mathbf{f} : X \rightarrow \mathbb{R}^m$ be Lipschitz near $\mathbf{x} \in X$ and directionally differentiable. \mathbf{f} is *lexicographically smooth* (or *l-smooth*) at \mathbf{x} if for any $q \in \mathbb{N}$ and any matrix $\mathbf{M} = [\mathbf{m}_1 \cdots \mathbf{m}_q] \in \mathbb{R}^{n \times q}$ the following functions are well-defined:

$$\begin{aligned} \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(0)} : \mathbb{R}^n &\rightarrow \mathbb{R}^m : \mathbf{d} \mapsto \mathbf{f}'(\mathbf{x}; \mathbf{d}), \\ \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(j)} : \mathbb{R}^n &\rightarrow \mathbb{R}^m : \mathbf{d} \mapsto [\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \mathbf{d}), \quad \forall j \in \{1, \dots, q\}. \end{aligned} \tag{2.2}$$

The function \mathbf{f} is said to be lexicographically smooth (*l-smooth*) on X if it is *l-smooth* at each point $\mathbf{x} \in X$.

The class of *l-smooth* functions includes all continuously and piecewise differentiable functions, all convex functions and is closed under composition. The elements of this homogenization sequence satisfy the following relations presented in Lemma 3 in [96]:

$$\begin{aligned} \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}(\tau \mathbf{d}) &= \tau \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}(\mathbf{d}), \quad \forall \mathbf{d} \in \mathbb{R}^n, \forall \tau \geq 0, \\ \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}(\mathbf{d} + \tau \mathbf{y}) &= \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}(\mathbf{d}) + \tau \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}(\mathbf{y}), \quad \forall \mathbf{d} \in \mathbb{R}^n, \\ &\quad \forall \mathbf{y} \in \text{span}\{\mathbf{m}_1, \dots, \mathbf{m}_k\}, \forall \tau \in \mathbb{R}, \end{aligned} \tag{2.3}$$

for all $k = 0, \dots, q$ and

$$\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}(\mathbf{d}) = \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k+1)}(\mathbf{d}) = \dots = \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(q)}(\mathbf{d}),$$

for all $\mathbf{d} \in \text{span}\{\mathbf{m}_1, \dots, \mathbf{m}_k\}$ and for all $k = 1, \dots, q-1$. Note that these relations imply that $\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}$ is linear on $\text{span}\{\mathbf{m}_1, \dots, \mathbf{m}_k\}$. In addition, the following property is also satisfied:

$$\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k-1)}(\mathbf{m}_k) = \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k+1)}(\mathbf{m}_k) = \dots = \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(q)}(\mathbf{m}_k), \quad \forall k \in \{1, \dots, q\}. \quad (2.4)$$

Definition 2.1.7. [96]. Let $\mathbf{f} : X \rightarrow \mathbb{R}^m$ be lexicographically smooth at $\mathbf{x} \in X$. Let $\zeta_k(\mathbf{f}, \mathbf{M}, \mathbf{x})$ be a Jacobian matrix of any linear function $\mathbf{c} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that

$$\mathbf{c}(\mathbf{d}) \equiv \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}(\mathbf{d}), \quad \mathbf{d} \in \text{span}\{\mathbf{m}_1, \dots, \mathbf{m}_k\}.$$

The Jacobian matrix $\zeta_k(\mathbf{f}, \mathbf{M}, \mathbf{x})$ is called an *l-k-derivative* of \mathbf{f} at \mathbf{x} along the sequence defined by the matrix \mathbf{M} . If $m = 1$, the column vector $\zeta_k^T(\rho, \mathbf{M}, \mathbf{x})$ is called the *l-k-gradient*.

Definition 2.1.8. [96]. The Jacobian matrix $\zeta(\mathbf{f}, \mathbf{M}, \mathbf{x})$ of the linear function $\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}$ with $k \geq \min\{r : \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(r)} \in L(\mathbb{R}^n; \mathbb{R}^m)\}$ is called the *lexicographic derivative (l-derivative)* of \mathbf{f} at \mathbf{x} along $\mathbf{M} \in \mathbb{R}^{n \times q}$. For $\mathbf{M} \in GL(n, \mathbb{R})$ denote the *l-derivative* by $\mathbf{J}_L \mathbf{f}(\mathbf{x}; \mathbf{M})$. Since \mathbf{M} is nonsingular *i.e.*, $\text{span}\{\mathbf{m}_1, \dots, \mathbf{m}_q\} = \mathbb{R}^n$, the *l-derivative* is given by $\mathbf{J}_L \mathbf{f}(\mathbf{x}; \mathbf{M}) = \mathbf{J} \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(n)}(\mathbf{0})$, the Jacobian of $\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(n)}$ at $\mathbf{0}$.

Nesterov shows that lexicographic derivatives exist whenever \mathbf{f} is *l-smooth* at \mathbf{x} [96]. If \mathbf{f} is (Fréchet) differentiable at \mathbf{x} , then $\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(k)}(\mathbf{d}) = \mathbf{J} \mathbf{f}(\mathbf{x}) \mathbf{d}$, for $k = 0, \dots, q$ and for any $\mathbf{M} \in \mathbb{R}^{n \times q}$.

Definition 2.1.9. [96]. Let the function $\mathbf{f} : X \in \mathbb{R}^n \rightarrow \mathbb{R}^m$ be *l-smooth* at $\mathbf{x} \in X$. The set

$$\partial_L \mathbf{f}(\mathbf{x}) \equiv \{\mathbf{J}_L \mathbf{f}(\mathbf{x}; \mathbf{M}) \in \mathbb{R}^{m \times n} : \mathbf{M} \in GL(n, \mathbb{R})\}$$

is called the *lexicographic subdifferential* of \mathbf{f} at \mathbf{x} .

For a scalar function f , it has been shown in [96] that $\partial_L f(\mathbf{x})$ is a subset of Clarke's generalized gradient ($\partial f(\mathbf{x})$), hence for any $\mathbf{M} \in GL(n, \mathbb{R})$ we have that $\mathbf{J}f_{\mathbf{x}, \mathbf{M}}^{(n)}(\mathbf{0}) \in \partial f(\mathbf{x})$. For vector-valued functions, the lexicographic subdifferential is no less useful than Clarke's generalized Jacobian for nonsmooth equation solving and optimization purposes because the lexicographic subdifferential is a subset of the plenary hull of the generalized Jacobian [72]. In addition, piecewise differentiable functions in the sense of Scholtes [119] are l -smooth and their l -derivatives are elements of the B-subdifferential [74].

The *lexicographic directional derivative* of \mathbf{f} (or *LD-derivative*) [74] at $\mathbf{x} \in X$ in the directions $\mathbf{M} \in \mathbb{R}^{n \times q}$ is

$$\mathbf{f}'(\mathbf{x}; \mathbf{M}) \equiv \left[\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(0)}(\mathbf{m}_1) \cdots \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(q-1)}(\mathbf{m}_q) \right] = \left[\mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(q)}(\mathbf{m}_1) \cdots \mathbf{f}_{\mathbf{x}, \mathbf{M}}^{(q)}(\mathbf{m}_q) \right].$$

This definition is particularly useful since first, for $\mathbf{M} \in GL(n, \mathbb{R})$ the LD-derivative and the l -derivative are related by $\mathbf{f}'(\mathbf{x}; \mathbf{M}) = \mathbf{J}_L \mathbf{f}(\mathbf{x}; \mathbf{M}) \mathbf{M}$, which is analogous to the relationship between the classical directional derivative and the Jacobian for smooth functions. However, \mathbf{M} does not have to be of full row rank to compute LD-derivatives, which can be extremely useful in the case of compositions.

Second, the chain rule for LD-derivatives has a simple and intuitive structure. Let $q \in \mathbb{N}$ and Y be an open subset of \mathbb{R}^p , let $\mathbf{g} : X \rightarrow Y$ and $\mathbf{f} : Y \rightarrow \mathbb{R}^m$ be l -smooth at $\mathbf{x} \in X$ and $\mathbf{g}(\mathbf{x})$, respectively. The LD-derivative of the l -smooth composition of $\mathbf{f} \circ \mathbf{g}$ at $\mathbf{x} \in X$ is given by the chain rule:

$$[\mathbf{f} \circ \mathbf{g}]'(\mathbf{x}; \mathbf{M}) = \mathbf{f}'(\mathbf{g}(\mathbf{x}); \mathbf{g}'(\mathbf{x}; \mathbf{M})). \quad (2.5)$$

Consider \mathbf{u} and \mathbf{v} to be lexicographically smooth functions with appropriate do-

mains and ranges. The sum and product rules follow from the chain rule [74]:

$$\begin{aligned}[\mathbf{u} + \mathbf{v}]'(\mathbf{x}; \mathbf{M}) &= \mathbf{u}'(\mathbf{x}; \mathbf{M}) + \mathbf{v}'(\mathbf{x}; \mathbf{M}), \\ [uv]'(\mathbf{x}; \mathbf{M}) &= v(\mathbf{x})u'(\mathbf{x}; \mathbf{M}) + u(\mathbf{x})v'(\mathbf{x}; \mathbf{M}).\end{aligned}$$

We can now revisit the examples where the sum and the chain rule result in overestimations.

Example 2.1.6. Consider $f(x) = g(x) + h(x)$ where $g(x) : x \mapsto \min(0, x)$ and $h(x) : x \mapsto \max(0, x)$. It is clear that $f(x) = x$ and therefore $f'(x) = \partial f(x) = 1$ for all $x \in \mathbb{R}$. Now consider $x = 0$ and $M > 0$. Then $f'(x; M) = f'(x)M = M$, $g'(0; M) = 0$ and $h'(0; M) = M$. Then $f'(x; M) = g'(x; M) + h'(x; M)$. If $M < 0$, $g'(0; M) = M$ and $h'(0; M) = 0$ and $f'(x; M) = g'(x; M) + h'(x; M)$.

Example 2.1.7. Consider $g(x) : x \mapsto \max(0, x)$, $h(x) : x \mapsto \min(0, x)$ and $f = [h \circ g]$. It is clear that $f(x) = 0$ for all $x \in \mathbb{R}$ and therefore $f'(x) = \partial f(x) = 0$ and for any M , $f'(x; M) = f'(x)M = 0$. Let $x = 0$ and $M > 0$. Then $g(0) = 0$, $g'(0; M) = M$ and $h'(g(0); g'(0; M)) = h'(0; M) = 0$. If $M < 0$, $g'(0; M) = 0$ and $h'(g(0); g'(0; M)) = h'(0; 0) = 0$. In either case, $f'(x; M) = h'(g(x); g'(x; M))$, which is the chain rule.

LD-derivatives are important in this thesis because bioprocesses can be modeled using dynamic flux balance analysis (DFBA), as explained in Chapter 1. In the remainder of this thesis, the concept of LD-derivatives will be used to develop an optimization framework for DFBA systems.

2.2 Algal biofuels

In recent years, due to climate change there has been an increased focus on the negative impacts of fossil fuels on the environment. As a result in March 2015, the United States pledged to cut its carbon emissions by 26-28% by 2025 [53]. This ambitious environmental objective was coupled with specific actions such as reducing

oil imports, increasing energy efficiency, and speeding up the development of biofuels [128]. Biofuels are a key component towards reducing emissions as liquid fuels are heavily used in the transportation sector and they currently account for 14% of global [64] and 27 % of United States [136] CO₂ emissions. As emissions are cut from fixed sources such as power plants, the transportation share of CO₂ emissions is expected to grow. In addition, although some of these emissions may be cut by using electric vehicles, liquid fuels will still be necessary for long-distance transportation as is the case of aviation. This is a consequence of liquid fuels having a much higher energy density compared to other energy carriers such as batteries or compressed gases (see Figure 2-1). The only way of reducing the impact of these emissions is by producing sustainable liquid fuels.

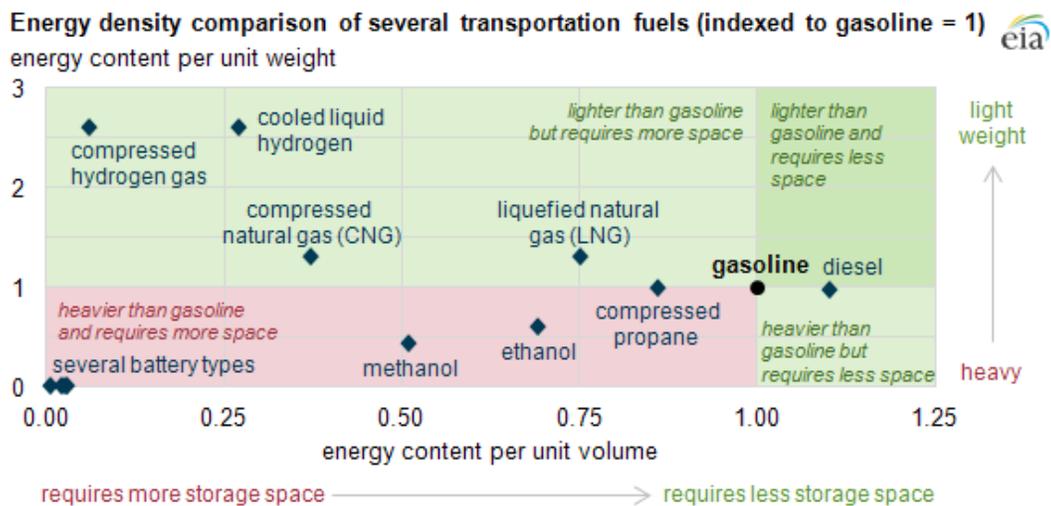


Figure 2-1: Energy density comparison of several transportation fuels (indexed to gasoline = 1). Figure obtained from [134].

Biofuels are fuels generated from biomass. First-generation biofuels are obtained from food crops, and had a production volume in the United States of approximately 50 billion litres in 2013, mainly corn ethanol. The production level of corn ethanol is expected to reach a maximum of approximately 55 billion litres per year, according to the United States Environmental Protection Agency and Energy Information Administration [123]. Despite providing improving domestic energy security, first-generation biofuels compete for food resources and are only slightly better than fossil fuels regard-

ing environmental impact. This has prompted research on second-generation biofuels which are obtained from waste biomass and show better figures regarding greenhouse gas emissions, carbon footprint, and environmental damage [84]. Second-generation biofuels represent a great opportunity because 349 million tons of sustainable waste biomass are produced per year just in the United States [2], and most of this biomass is wasted.

Waste biomass can be converted into biofuels with the help of microorganisms through microbial conversion processes or microbial biomass production. Microbial conversion relies on fermentation or anaerobic digestion to obtain fuels from the secretions of microorganisms such as bioethanol or biogas. Meanwhile, biofuels relying on microbial biomass production are obtained from the lipids accumulated by microorganisms growing on waste biomass and/or sunlight to produce biodiesel. The remaining biomass can be digested anaerobically or be regarded as waste [58]. Biodiesel is attractive because it has a higher energy density than bioethanol. Three types of microorganisms are used for microbial biomass production: bacteria, fungi (including higher fungi), and microalgae. Preferred characteristics of the microorganisms are high specific growth rate, high lipids to biomass yield, high cell density, ability to use complex substrates, affinity to substrate, and low nutrient requirements [58].

Microalgae are attractive for biofuels production from sunlight energy because some strains naturally accumulate up to 50% dry weight in lipids [141]. In addition, algae do not compete for food resources as they can be grown on wastewater and/or seawater [26], and they are up to one order of magnitude more efficient than higher-order terrestrial plants in capturing sunlight [141, 25]. In addition, algal biofuels have reduced CO₂ emissions compared to fossil fuels, and can become carbon neutral if all energy inputs to the supply chain are carbon neutral. Despite all these advantages, algal biofuels remain to be commercialized due to their high prices. For example, in 2013 the Department of Defense paid \$150 per gallon for 1,500 gallons of jet fuel when petroleum-based jet fuel was only \$2.88 per gallon [133]. Prices remain high because a low cost production method that obtains acceptable algal biomass and lipids yields remains to be found.

Oleaginous yeasts are also attractive for biofuels production as they can convert lignocellulosic sugars into lipids. Some examples of oleaginous yeast strains include *Cryptococcus albidus*, *Lipomyces starkeyi*, *Rhodotorula glutinis*, *Trichosporon pullulans*, and *Yarrowia Lipolytica* which accumulate up to 65, 63, 72, 65, and 36 % lipids, respectively [105, 11]. Although some microalgae are able to grow mixotrophically, yeasts are able to metabolize a wider range of carbon substrates compared to algae. In fact, some yeast strains are able to metabolize both glucose and xylose, making them good candidates for lignocellulosic waste conversion to biofuels [45].

Algae and yeasts can be cultivated in open pond systems or closed photobioreactors. Closed photobioreactors have been used successfully to produce high-value specialty chemicals [36], but these systems incur high capital and operating costs for the production of commodities such as biofuels [4]. On the other hand, open pond lipid yields are insufficient because monocultures are vulnerable to invasion and predation by other algae species, bacterial or fungal infection. Oleaginous yeasts that thrive under low pH and low temperature conditions have been successfully cultivated in open ponds [115], but most oleaginous yeasts are not extremophiles. In this case, culture resilience and stability are critical. Synthetic consortia can be designed to fill ecological niches which would otherwise be filled by invading species. Design of such synthetic consortia has been discussed in Kazamia et al. [70] at a qualitative level, and a quantitative approach has been proposed in Höffner and Barton [58]. In addition, algae in open pond cultures are carbon limited due to the low atmospheric CO₂ concentration and yeasts can become O₂ limited [21]. The carbon limitation has restricted the locations where algal ponds can be economically feasible, because the use of CO₂-rich flue gas is only possible in the vicinity of power plants [9]. An alternative approach to cultivating monocultures of yeast and algae is to grow them together and benefit from their symbiotic interactions. Examples of this approach have been tested at lab scale [21, 116, 108, 145, 102, 82, 148]. The introduction of yeast enables lignocellulosic sugars, which cannot be metabolized by most microalgae, to be digested and can increase algal biomass by transforming part of these carbon sources into CO₂. At the same time, yeast can benefit from the O₂ produced by microal-

gae and increase lipids production. In addition, both species together fill available ecological niches to protect against invasion [70]. This alternative strategy promotes installing algal/fungal ponds near farms, where significant quantities of agricultural waste are generated, but no flue gas is available, and transform these wastes into lipids first, and then biodiesel.

The quantitative approach proposed in Höffner and Barton [58] to design synthetic consortia requires good bioprocess models. The modelling of microbial consortia in open ponds is challenging because complex phenomena such as growth under multiple substrate and nutrient limitations, symbiotic relationships, and day/night transitions are present. These phenomena result in microorganisms switching between different growth modes over the course of the day, which are complicated to model as explained in Chapter 1. Traditional bioprocess modelling relies on unstructured models (e.g. Monod, Tessier, Moser, Blackman equations), which are derived for microorganisms in a single growth mode [122]. To model microbial growth in an open pond using unstructured models, all growth modes of the different microorganisms need to be identified, their constants obtained experimentally, and rules for transitions from one growth mode to the next derived. This makes the modeling of open ponds using unstructured models intractable. These limitations are addressed by flux balance analysis [137, 98] by considering genome-scale metabolic network reconstructions of all microorganisms in the culture to predict growth and exchange fluxes rates.

The work in Chapters 4 and 5 is aimed at creating a reliable process model for a raceway pond. This model can be optimized to reduce the costs of cultivating biomass and obtain cheaper biofuels.

Chapter 3

DFBALab: A fast and reliable MATLAB code for Dynamic Flux Balance Analysis

This chapter reproduces the article [44]. It introduces DFBALab, a fast and reliable MATLAB code for DFBA simulations.

The acceleration in the process of genome sequencing in recent years has increased the availability of genome-scale metabolic network reconstructions for a variety of species (see for example [126]). These genome-based networks can be used within the framework of flux balance analysis (FBA) to predict steady-state growth and uptake rates accurately [98]. Dynamic flux balance analysis (DFBA) enables the simulation of dynamic biological systems by assuming organisms reach steady state rapidly in response to changes in the extracellular environment [125]. Then, the rates predicted by FBA are used to update the extracellular environment. There exist three approaches to simulate DFBA models: the static optimization approach (SOA) [87], the dynamic optimization approach [87] (DOA), and the direct approach (DA). The static optimization approach uses the Euler forward method, solving the embedded LPs at each time step. Since most DFBA models are stiff, small time steps are required for stability, making this approach computationally expensive. Meanwhile, the DOA approach discretizes the time horizon and optimizes simultaneously over

the entire time period of interest by solving a nonlinear programming problem (NLP) (see [104] for an example of this approach). The dimension of this NLP increases with time discretization, therefore it is limited to small-scale metabolic models [59]. Finally, a DA has been proposed recently by including the LP solver in the right-hand side evaluator for the ordinary differential equations (ODEs) and taking advantage of reliable implicit ODE integrators with adaptive step size for error control. At present, the DOA is rarely used due to the intractability of the resulting NLP. DFBA can be easily performed on MATLAB using the constraint-based reconstruction and analysis (COBRA) toolbox [118, 98], which implements the SOA. Recently, the DA has been implemented by Hanly and Henson [51], Mao and Verwoerd in the ORCA toolbox [90], Zhuang *et al.* in the dynamic multispecies metabolic modeling (DyMMM) framework [150, 149], and others. A comprehensive list of DFBA implementations can be found in Table I of [59]. COBRA, DyMMM and ORCA codes are available on the Web. Of these, only DyMMM allows community simulations. Since ORCA and DyMMM are extremely similar, only COBRA and DyMMM were implemented in the case studies presented.

These implementations present several shortcomings. The COBRA Toolbox uses a fixed time step and does not take advantage of the high quality built-in integrators provided by MATLAB. Simulation stability and accuracy are closely linked to a uniformly small step size which can greatly increase simulation time. It can fail if the extracellular conditions are close to the FBA model becoming infeasible. In addition, it uses a simple exchange flux bounding scheme that does not allow the implementation of Michaelis-Menten kinetics or other more complex dynamic behaviors such as day/night shifts for photosynthetic organisms or system feed and discharge rates. It does not allow community simulations.

The ORCA toolbox and the DyMMM framework use the MATLAB built-in integrators. ORCA simulates monocultures only, whereas DyMMM can simulate cocultures. The ORCA toolbox allows the implementation of Michaelis-Menten and Hill kinetics only, whereas DyMMM provides the flexibility to implement more complex dynamics such as day/night shifts for photosynthetic organisms or system feed and

discharge rates. Both attempt to carry on with simulations when the FBA model is infeasible by setting the growth rate and exchange fluxes equal to zero and displaying a death phase message. This message may be displayed prematurely when the system is still feasible and introduces discontinuous behavior.

None of these implementations (COBRA, ORCA, and DyMMM) accounts for the solution of a linear program (LP) being a nonsingleton set. Therefore, exchange fluxes are not necessarily unique and the dynamic system is not well-defined. Nonunique optimal fluxes have been discussed elsewhere in [86] and [51]. If no effort is made to obtain unique fluxes, different integrators could yield different results and small changes on the initial conditions or the tolerances could lead to dramatic and unpredictable simulation changes.

Höffner *et al.* have designed a fast and reliable community simulator that has the flexibility of implementing complex dynamics, does not fail due to LP infeasibilities, identifies precisely when a system becomes infeasible, and performs lexicographic optimization to render unique exchange fluxes [59, 56]. In particular, it avoids numerical failure by reformulating the LP as an algebraic system and integrating an index-1 differential-algebraic equation (DAE) system. Despite these advantages, this simulator has not been widely used due to being coded in FORTRAN. In this chapter, we implement the LP feasibility problem combined with lexicographic optimization in our Dynamic Flux Balance Analysis laboratory (DFBALab), a MATLAB code that performs fast, reliable and flexible community simulations.

3.1 Implementation

DFBALab provides a solution to three major difficulties in existing implementations: nonunique exchange fluxes in the solution vector of an LP, the LP becoming infeasible when evaluating the ODE right-hand side close to the boundary of feasibility, and the computational expense associated with solving the FBA LP. DFBALab implements lexicographic optimization to obtain unique exchange fluxes [56], uses the LP feasibility problem to avoid obtaining infeasible LPs while running the simulation,

and uses LP basis information to reformulate the ODE system with LPs embedded into a sequence of differential-algebraic equation (DAE) systems in time. DFBalab runs using the commercial linear program solvers CPLEX [28] and Gurobi [50], and is compatible with the COBRA toolbox model format.

3.1.1 Lexicographic optimization

Dynamic flux balance analysis is defined in the following way. Consider a vector \mathbf{x}_0 containing the initial concentrations of metabolites and biomass in a culture and assume there are n_s microbial species in the culture. Given some uptake and production rates of metabolites for each species (exchange fluxes), feed and discharge rates from the culture, mass transfer rates, and other dynamic processes, a rate of change function \mathbf{f} can be obtained for each of the components of \mathbf{x}_0 . The function \mathbf{f} can then be integrated to find the concentration profiles with respect to time, $\mathbf{x}(t)$. Consider that each species k has n_h^k exchange fluxes and define the linear maps $\mathbf{B}^k : \mathbb{R}^{n_r^k} \rightarrow \mathbb{R}^{n_h^k}$ which obtain the exchange fluxes (*e.g.* biomass production rate, O_2 consumption rate, ethanol production rate, etc.) from the n_r^k metabolic fluxes. Formally, given the nonempty open set $D_x \subset \mathbb{R}^{n_x}$, $\mathbf{f} : [t_0, t_f] \times D_x \times \mathbb{R}^{n_h^1} \times \dots \times \mathbb{R}^{n_h^{n_s}} \rightarrow \mathbb{R}^{n_x}$, $\mathbf{v}^k : D_x \rightarrow \mathbb{R}^{n_r^k}$, $\mathbf{v}_{LB}^k : D_x \rightarrow \mathbb{R}^{n_r^k}$, $\mathbf{v}_{UB}^k : D_x \rightarrow \mathbb{R}^{n_r^k}$ for $k = 1, \dots, n_s$, and $\mathbf{x} : [t_0, t_f] \rightarrow \mathbb{R}^{n_x}$:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{B}^1(\mathbf{v}^1(\mathbf{x}(t))), \dots, \mathbf{B}^{n_s}(\mathbf{v}^{n_s}(\mathbf{x}(t))))), \quad \forall t \in (t_0, t_f], \quad (3.1)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0,$$

where \mathbf{v}^k is an element of the solution set of the flux balance model of species k :

$$\begin{aligned} \mathbf{v}^k(\mathbf{x}(t)) \in \arg \max_{\mathbf{v} \in \mathbb{R}^{n_r^k}} (\mathbf{c}^k)^\top \mathbf{v}, \\ \text{s.t. } \mathbf{S}^k \mathbf{v} = \mathbf{0}, \\ \mathbf{v}_{UB}^k(\mathbf{x}(t)) \geq \mathbf{v} \geq \mathbf{v}_{LB}^k(\mathbf{x}(t)), \end{aligned} \quad (3.2)$$

where $\mathbf{S}^k \in \mathbb{R}^{n_q^k \times n_r^k}$ is the stoichiometry matrix, $\mathbf{c}^k \in \mathbb{R}^{n_r^k}$ is a vector of zeroes and ones with ones only in positions of growth fluxes, and $\mathbf{v}_{LB}^k, \mathbf{v}_{UB}^k$ are lower and upper bounds as functions of the extracellular concentrations. This definition of DFBA has a serious problem: the solution set of the LP (3.2) can be nonunique (*e.g.* different flux distributions \mathbf{v}^k can attain the maximum growth rate) and it is not clear which flux distribution should the LP solver choose to carry-on with the integration. This behavior is illustrated in Example 1.0.1.

In the rest of this document, we will work with the *standard form* LP of (3.2). Let $\mathbf{A}^k \in \mathbb{R}^{n_m^k \times n_v^k}$, $\mathbf{c}^k \in \mathbb{R}^{n_v^k}$, $\boldsymbol{\beta} \in \mathbb{R}^{n_m^k}$:

$$\begin{aligned} \min_{\mathbf{v} \in \mathbb{R}^{n_v^k}} & (\mathbf{c}^k)^T \mathbf{v}, \\ \text{s.t.} & \mathbf{A}^k \mathbf{v} = \boldsymbol{\beta}, \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{3.3}$$

It is well known that any linear program can be rewritten in standard form [12]. The information of \mathbf{v}_{LB}^k and \mathbf{v}_{UB}^k is now in the right-hand side vector $\boldsymbol{\beta}$. Then, for each species k , let $\mathbf{b}^k : D_x \rightarrow \mathbb{R}^{n_m^k}$.

Harwood and coworkers [56] use lexicographic optimization to render unique exchange fluxes. Lexicographic optimization works in the following way. First, it orders a number of objectives in a priority list. The highest priority objective is optimized first; then its optimum value is added as a constraint and the next objective in priority is optimized, and so on. Lexicographic optimization can be implemented in DFBA systems: the first objective is maximization of biomass; then all other exchange fluxes that appear in the right-hand side of (3.1) are added to the priority list. Note that the choice of the objective functions and their ordering are part of the model description and must be provided by the user. Although LPs don't necessarily have a unique flux distribution that attains the optimal objective function value, they do have a unique optimal objective function value. This optimal objective function value changes continuously with changes in $\mathbf{v}_{LB}^k, \mathbf{v}_{UB}^k$. By making all the exchange fluxes that appear in the right-hand side of (3.1) optimization objectives ordered by priority,

unique exchange fluxes are obtained, these exchange fluxes change continuously with respect to time and the integrator is able to carry-on integration reliably.

Let $\mathbf{h}^k : D_x \rightarrow \mathbb{R}^{n_h^k}$, then:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(t, \mathbf{x}(t), \mathbf{h}^1(\mathbf{x}(t)), \dots, \mathbf{h}^{n_s}(\mathbf{x}(t))), \quad \forall t \in (t_0, t_f], \\ \mathbf{x}(t_0) &= \mathbf{x}_0.\end{aligned}\tag{3.4}$$

The function $\mathbf{h}^k = [h_1^k \dots h_{n_h^k}^k]^\top$ depends on the solution of a lexicographic LP (LLP):

$$\begin{aligned}h_1^k(\mathbf{x}(t)) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v^k}} (\mathbf{c}_1^k)^\top \mathbf{v}, \\ \text{s.t. } \mathbf{A}^k \mathbf{v} &= \mathbf{b}^k(\mathbf{x}(t)), \\ \mathbf{v} &\geq \mathbf{0},\end{aligned}\tag{3.5}$$

and for $2 \leq i \leq n_h^k$

$$\begin{aligned}h_i^k(\mathbf{x}(t)) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v^k}} (\mathbf{c}_i^k)^\top \mathbf{v}, \\ \text{s.t. } \begin{bmatrix} \mathbf{A}^k \\ (\mathbf{c}_1^k)^\top \\ \vdots \\ (\mathbf{c}_{i-1}^k)^\top \end{bmatrix} \mathbf{v} &= \begin{bmatrix} \mathbf{b}^k(\mathbf{x}(t)) \\ h_1^k(\mathbf{x}(t)) \\ \vdots \\ h_{i-1}^k(\mathbf{x}(t)) \end{bmatrix}, \\ \mathbf{v} &\geq \mathbf{0},\end{aligned}\tag{3.6}$$

where $\mathbf{c}_i^k \in \mathbb{R}^{n_v^k}$ for $i = 1, \dots, n_h^k$. A more compact version of (7.2) and (7.3) can be obtained by defining the lexicographic minimization operator lex min . Let the columns of $\mathbf{C}^k \in \mathbb{R}^{n_v^k \times n_h^k}$ be the vectors \mathbf{c}_i^k for $i = 1, \dots, n_h^k$. Then,

$$\begin{aligned}\mathbf{h}^k(\mathbf{x}(t)) &= \text{lex min}_{\mathbf{v} \in \mathbb{R}^{n_v^k}} (\mathbf{C}^k)^\top \mathbf{v}, \\ \text{s.t. } \mathbf{A}^k \mathbf{v} &= \mathbf{b}^k(\mathbf{x}(t)), \\ \mathbf{v} &\geq \mathbf{0}.\end{aligned}\tag{3.7}$$

Harwood *et al.* [56] present an efficient algorithm to compute a basis that contains optimal bases for all LPs in the priority list. This algorithm is critical to reformulate the ODE system with LPs embedded as a sequence of DAE systems in time.

3.1.2 LP Feasibility Problem

A major problem for DFBA simulators is that the LP in (3.2) may become infeasible as time progresses. There are two situations where the LP may become infeasible:

1. The problem is truly infeasible and the solution cannot be continued: in this case the integration should be terminated.
2. The problem is not infeasible but the LP becomes infeasible while the numerical integrator performs various operations to take a time step in (3.1): in this case the DFBA simulator in COBRA may fail to continue the simulation and ORCA and DyMMM will erroneously display death phase messages introducing a discontinuity in the right-hand side of (3.1). In particular, the MATLAB's built-in integrators will have a hard-time obtaining reliable right-hand side information as the system changes abruptly from being defined by the solution to (3.2), to being defined by an artificial solution that sets growth rates and exchange fluxes equal to zero.

In this chapter we use the LP feasibility problem [12] combined with lexicographic optimization to generate an extended dynamic system for which the LP always has a solution. An LP feasibility problem finds a feasible point or identifies an LP as infeasible. It has two main characteristics: it is always feasible and its optimal objective function value is zero if and only if the original LP is feasible. Several different versions of the LP feasibility problem can be constructed by adding some slack variables to the constraints. For the LP formulation in (3.2), the following is an LP feasibility

problem:

$$\begin{aligned}
& \min_{\substack{\mathbf{v} \in \mathbb{R}^{n_r^k}, \\ \mathbf{s}_+, \mathbf{s}_- \in \mathbb{R}^{n_q^k}}} \sum_{i=1}^{n_q^k} s_{+i} + s_{-i}, \\
& \text{s.t. } \mathbf{S}^k \mathbf{v} + \mathbf{s}_+ - \mathbf{s}_- = \mathbf{0}, \\
& \mathbf{v}_{UB}^k(\mathbf{x}(t)) \geq \mathbf{v} \geq \mathbf{v}_{LB}^k(\mathbf{x}(t)), \\
& \mathbf{s}_+ \geq \mathbf{0}, \mathbf{s}_- \geq \mathbf{0}.
\end{aligned} \tag{3.8}$$

Let \mathbf{S}_i be the i^{th} row of \mathbf{S} . When an LP is constructed in this form, a feasible solution is obtained by finding a \mathbf{v} such that $\mathbf{v}_{UB}^k(\mathbf{x}(t)) \geq \mathbf{v} \geq \mathbf{v}_{LB}^k(\mathbf{x}(t))$ and then letting $s_{+i} = -\mathbf{S}_i^k \mathbf{v}$ and $s_{-i} = 0$ if $\mathbf{S}_i^k \mathbf{v} < 0$, or $s_{-i} = -\mathbf{S}_i^k \mathbf{v}$ and $s_{+i} = 0$ otherwise. DFBAlab transforms LP (3.2) to standard form and then obtains the LP feasibility problem for an LP in standard form ([12]); however, the principles are the same. Any LP in standard form (3.3) can be transformed such that $\boldsymbol{\beta} \geq \mathbf{0}$ by multiplying some equality constraints by -1. Then, the LP feasibility problem will have the following general structure [12]:

$$\begin{aligned}
& \min_{\mathbf{v} \in \mathbb{R}^{n_v^k}, \mathbf{s} \in \mathbb{R}^{n_m^k}} \sum_{i=1}^{n_m^k} s_i, \\
& \text{s.t. } \mathbf{A}^k \mathbf{v} + \mathbf{s} = \boldsymbol{\beta}, \\
& \mathbf{v} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0}.
\end{aligned} \tag{3.9}$$

When an LP is constructed in this form, a feasible solution is obtained by setting $\mathbf{s} = \boldsymbol{\beta}$ and $\mathbf{v} = \mathbf{0}$.

DFBAlab uses the LP feasibility problem (3.9) instead of (3.2) to find the growth rates and exchange fluxes for each species in the culture. It sets the feasibility cost vector as the top priority objective in the lexicographic optimization scheme. Then, the second-priority LP maximizes biomass and the subsequent lower-priority LPs obtain unique exchange fluxes. The order of the exchange fluxes in the priority list is user-defined. The priority list order is fixed throughout the simulation. This order has

to be defined carefully or unrealistic simulation results may be obtained (as illustrated in Example 3.2.2). This approach has the following advantages:

1. The dynamic system in (3.1) is defined for all simulation time.
2. The integrator does not encounter infeasible LPs while taking a step and is able to obtain reliable right-hand side information speeding up the integration process.
3. The objective function value of (3.8) provides a distance from feasibility and can be integrated providing a penalty function that can be useful for optimization purposes. Only trajectories with penalty function value equal to zero (within some tolerance ϵ) are feasible.

3.1.3 Reformulation as a DAE system

DFBALab uses the strategies described in [56, 59] to transform the FBA problem into a sequence of DAE systems in time. This is done by observing that for LPs in standard form, the reduced costs remain invariant for perturbations in the right-hand side [12]. Therefore, once an optimal basis is obtained for an LLP, this basis will remain optimal as long as it is feasible [12, 56]. Consider LLP (3.7) and an optimal basis B^k . This basis will remain feasible as long as $(\mathbf{A}_{B^k}^k)^{-1}\mathbf{b}^k(\mathbf{x}(t)) \geq \mathbf{0}$. In addition, $\mathbf{h}^k(\mathbf{x}(t)) = (\mathbf{C}_{B^k}^k)^\top(\mathbf{A}_{B^k}^k)^{-1}\mathbf{b}^k(\mathbf{x}(t))$. The feasibility condition with the algebraic equation enable the reformulation of the DFBA problem as a sequence of DAE systems in time where the transition from one DAE system to the next is given by the feasibility conditions of the basis. More details on the resulting algorithm can be found in [56, 59]. For these ideas to work, \mathbf{A}^k must be full row rank for all k . DFBALab uses QR factorization to obtain a full row rank system of equations analogous to $\mathbf{A}^k\mathbf{v} = \mathbf{b}^k(\mathbf{x}(t))$.

3.2 Results and Discussion

The following examples demonstrate the reliability and speed of DFBAlab compared to existing implementations of the SOA and DA. SOA is represented by the COBRA dFBA implementation and DA by the DyMMM implementation. In the first example, a monoculture of *E. coli* is simulated with all three methods. In the second example, a coculture of algae and yeast is simulated using DFBAlab and DyMMM. In the third example, this same coculture is simulated considering the pH balance. Finally, the last example shows how DFBAlab running time increases linearly with the number of FBA models in the system. All running times are for a 3.20 GHz Intel® Xeon® CPU in MATLAB 7.12 (R2011a), Windows 7 64-bit operating system using LP solver CPLEX. All running times are for the integration process only (preprocessing times are not reported). DFBA models are usually stiff; therefore, ode15s, MATLAB's integrator for stiff systems, was used for all simulations.

Example 3.2.1. This is Example 6.2 in [56] which is based on [51]. Here we compare the performance of COBRA, DyMMM and DFBAlab simulating an *E. coli* monoculture. The metabolic network reconstruction used was iJR904 published in [107]. This metabolic model contains 1075 reactions and 761 metabolites. Initial conditions were 0.03 g/L of inoculum, 15.5 g/L of glucose and 8 g/L of xylose. Oxygen concentration was kept constant at 0.24 mmol/L. Michaelis-Menten expressions with inhibition terms were implemented to bound the uptake of glucose, xylose and oxygen using the parameters presented in Table I and Equations (3), (4) and (5) in [51]. DFBAlab obtained unique fluxes by minimizing ethanol production, and then glucose and xylose consumption, after maximizing biomass, using lexicographic optimization. The COBRA simulator performed poorly. Since COBRA does not have the flexibility to implement Michaelis-Menten expressions, the simulation results were incorrect. In addition, the fixed step size slowed down the integration process. Non-negativity constraints for all states variables were enforced in both DyMMM and DFBAlab, by using the 'Nonnegative' option. DyMMM and DFBAlab obtained the same concentration profiles presented in Figure 3-1. DyMMM has a good performance recovering

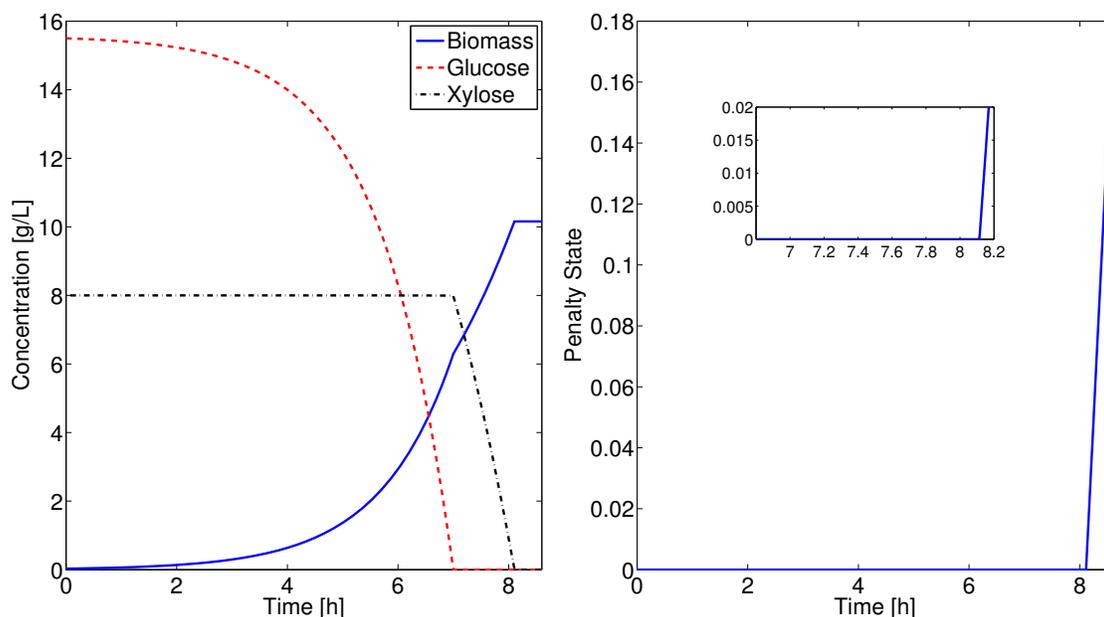


Figure 3-1: Concentration profiles (left) and DFBA lab penalty function (right) of Example 3.2.1. The penalty function shows how the simulation becomes infeasible after approximately 8.1 hours. Simulation times: DyMMM = 6.6 seconds, DFBA lab = 1.4 seconds.

from a frequent failure point occurring when growth switches from glucose-based to xylose-based. DFBA lab performs faster than DyMMM despite the four additional LPs being solved to perform lexicographic optimization, obtaining at least the same level of accuracy. Finally, the penalty function indicates that the system becomes infeasible after approximately 8.1 hours (Figure 3-1).

Example 3.2.2. This is an example from [58] of a coculture of the microalgae *Chlamydomonas reinhardtii* and *Saccharomyces cerevisiae* (yeast) in a continuous stirred-tank reactor (CSTR) reactor. The genome-scale metabolic network reconstructions used were iRC1080, comprising 2191 reactions and 1706 metabolites from [19], and iND750, comprising 1266 reactions and 1061 metabolites from [31], for algae and yeast, respectively. In this simulation, yeast consumes glucose to produce CO_2 while algae consumes mainly CO_2 to produce O_2 during the day, and acetate to produce CO_2 during the night. The dynamic mass balance equations of the extracellular

environment for this system are:

$$\dot{y}^i(t) = \mu^i(\mathbf{x}(t))y^i(t) - \frac{F_{out}y^i(t)}{V}, \quad (3.10)$$

$$\dot{s}(t) = \frac{F_{in}s_0 - F_{out}s(t)}{V} + MT_s(\mathbf{x}(t)) + \sum_i (v_{s_p}^i(\mathbf{x}(t)) - v_{s_c}^i(\mathbf{x}(t)))y^i(t), \quad (3.11)$$

for $i = Y, A$, for $s = g, o, c, e, a$,

where y^i , g , o , c , e , and a correspond to the concentrations of biomass of species i , glucose, oxygen, carbon dioxide, ethanol and acetate, respectively. The superscripts Y, A refer to yeast and algae, $\mathbf{x} = [y^Y \ y^A \ g \ o \ c \ e \ a]$, μ^i is the growth rate of species i , $v_{s_c}^i$ and $v_{s_p}^i$ are the consumption and production rates of substrate s for species i determined through lexicographic optimization, s_0 is the concentration of s in the feed, F_{in} and F_{out} are the inlet and outlet flows, V is the volume of the system, and MT_s is the mass transfer rate of s given by the following expression:

$$MT_s(\mathbf{x}(t)) = \begin{cases} (k_{sL}\theta) \left(\frac{s^{(g)}}{K_{Hs}} - s(t) \right) & \text{for } s = o, c, \\ 0 & \text{for } s = g, e, a, \end{cases} \quad (3.12)$$

where K_{Hs} refers to Henry's constant of component s at 25 °C, $k_{sL}\theta$ is the mass transfer coefficient for component s (from [18]), and $s^{(g)}$ is the concentration of s in the atmosphere. The maximum concentration of oxygen and carbon dioxide in the culture is bounded by Henry's constant:

$$s(t) \leq K_{Hs}, \quad \forall t \in [t_0, t_f] \quad \text{for } s = o, c. \quad (3.13)$$

Initial concentrations and other parameters are presented in Table 3.1. The uptake kinetics are bounded above by the Michaelis-Menten expression:

$$v_s^{i,UB}(s(t)) = v_{s,max}^i \frac{s(t)}{K_s^i + s(t)}, \quad (3.14)$$

for $i = Y, A$ and $s = a, o, c$ with $v_{s,max}^i$ and K_s^i obtained from [147], [6] and [144] for acetate, carbon dioxide and oxygen. Production of oxygen by algae, ethanol by yeast,

Table 3.1: Initial concentrations and parameters of Example 3.2.2. Simulation 1 used the lexicographic objectives presented in Table 3.2, while for Simulation 2 objective 4 for algae was inverted.

Variable	Simulation 1	Simulation 2	Parameters
y_0^Y	1.10	0.71	gDW/L
y_0^A	1.86	1.80	gDW/L
g_0	1.40 E^{-2}	2.28 E^{-2}	mmol/L
o_0	6.53 E^{-4}	5.57 E^{-4}	mmol/L
c_0	1.06	1.03	mmol/L
e_0	8.21	17.32	mmol/L
a_0	2.39 E^{-2}	2.48 E^{-2}	mmol/L

Table 3.2: Priority list order for the lexicographic linear programs in Example 3.2.2.

	Yeast	Algae
1	Minimize slacks of feasibility LP	Minimize slacks of feasibility LP
2	Maximize biomass production	Maximize biomass production
3	Minimize glucose consumption	Maximize acetate consumption
4	Minimize O ₂ consumption	Minimize consumption and maximize production of O ₂
5	Maximize CO ₂ production	Maximize CO ₂ consumption
6	Maximize ethanol production	

and carbon dioxide by algae and yeast were not bounded.

In addition to the extracellular concentrations, algae growth is affected by light availability because it is a photosynthetic organism. Day and night shifts were simulated using the following surface light function:

$$I_0(t) = 28 \frac{\max\left(\sin^2\left(\frac{2\pi t}{48}\right), \sin^2\left(\frac{10\pi}{48}\right)\right) - \sin^2\left(\frac{10\pi}{48}\right)}{1 - \sin^2\left(\frac{10\pi}{48}\right)} (=) \frac{\text{mmol photons}}{gDW \times h}. \quad (3.15)$$

This light function simulates daylight from 5:00 to 19:00. The prefactor was obtained from [43]. The Beer-Lambert law was used to average the light available to algae cells considering that higher biomass densities block light and deeper sections of the pond receive less sunlight:

$$I_a(t, \mathbf{x}(t)) = I_0(t) \frac{1 - \exp(-LK_e(\mathbf{x}(t)))}{LK_e(\mathbf{x}(t))}, \quad (=) \frac{\text{mmol photons}}{gDW \times h}, \quad (3.16)$$

where $K_e(\mathbf{x}(t))$ is a linear function of the concentration of biomass in the culture and

L is the pond depth [144]. Concentration variations of biomass for different pond depths were neglected.

This complex community simulation cannot be carried out using the DFBA simulator in COBRA. Non-negativity constraints were enforced for all state variables in both, DyMMM and DFBAlab, by using the ‘Nonnegative’ option. After more than 10,000 seconds of running time using MATLAB implicit integrator ode15s, the simulation on DyMMM was stopped. Using explicit integrator ode45 instead, DyMMM took more than 3900 seconds to simulate one hour of the cyclic steady-state of this coculture and the results are inaccurate. This is expected because explicit integrators can calculate new steps as long as they are able to evaluate the right-hand side of the ODE. The results obtained by DyMMM using ode45 are inaccurate because explicit integrators should not be used for stiff systems, and the right-hand side is nonunique. In Figure 3-2, it can be seen that the acetate curve presents several points of nonsmoothness which are expected in systems with nonunique fluxes. Numerical integrators are unable to handle these systems as they encounter discontinuous exchange fluxes when decreasing step-size. Therefore, computation time is excessive and the results are incorrect. This shortcoming is addressed by DFBAlab using six lexicographic optimizations for yeast and five for algae. It took only 16.4 seconds to simulate accurately 24 hours of this coculture using the lexicographic objectives shown in Table 3.2, and 15.3 seconds to simulate this same system with the negative of Objective 4 for algae. Simulation results can be seen in Figure 3-3.

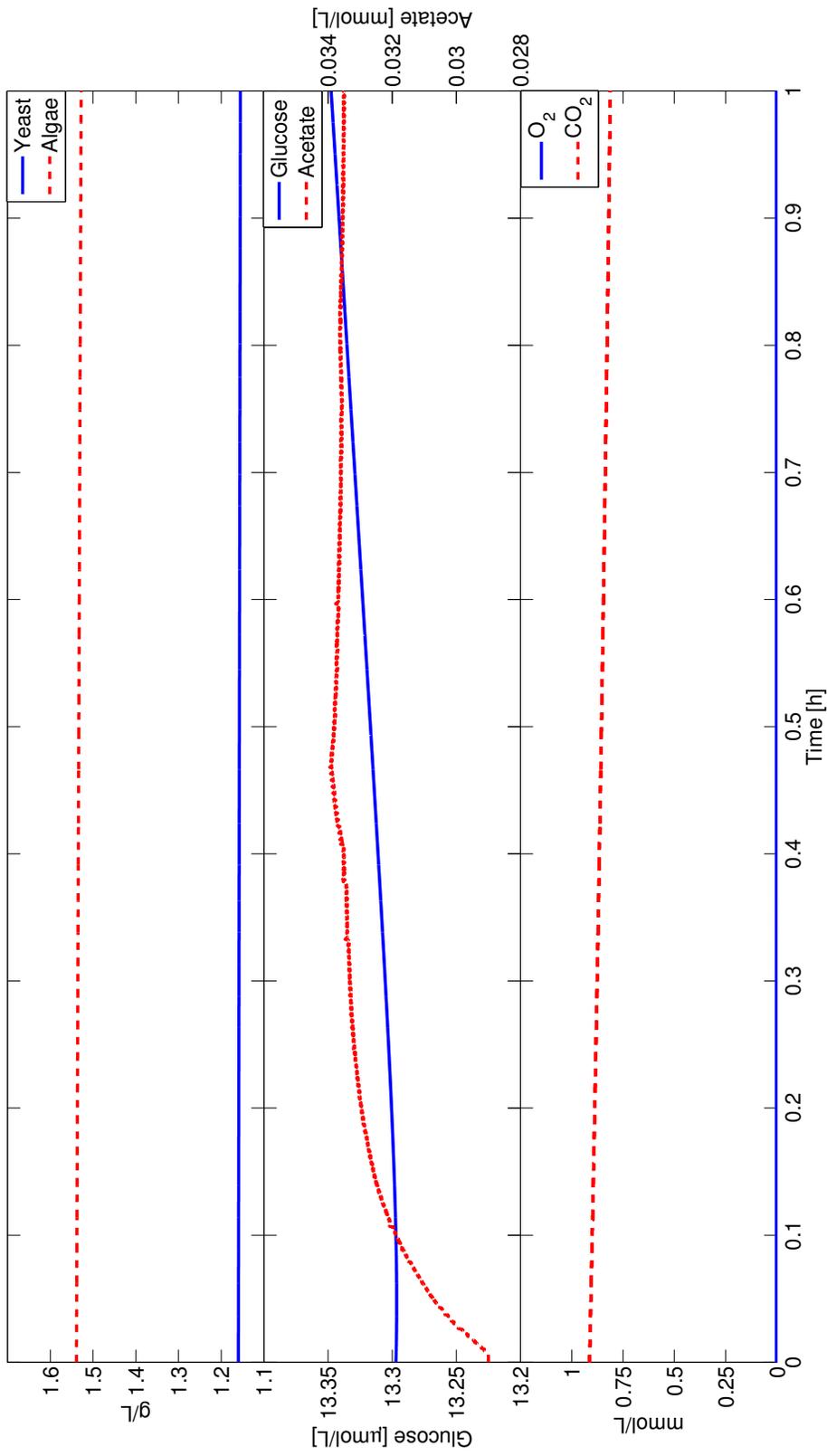


Figure 3-2: DyMMM simulation results of Example 3.2.2. DyMMM is unable to simulate Example 3.2.2. Computation time for one hour of simulation was of more than 3900 seconds using MATLAB explicit integrator ode45. In addition, the acetate curve has several points of nonsmoothness that can be explained by the presence of nonunique fluxes. Numerical integrators are unable to integrate these kinds of systems.

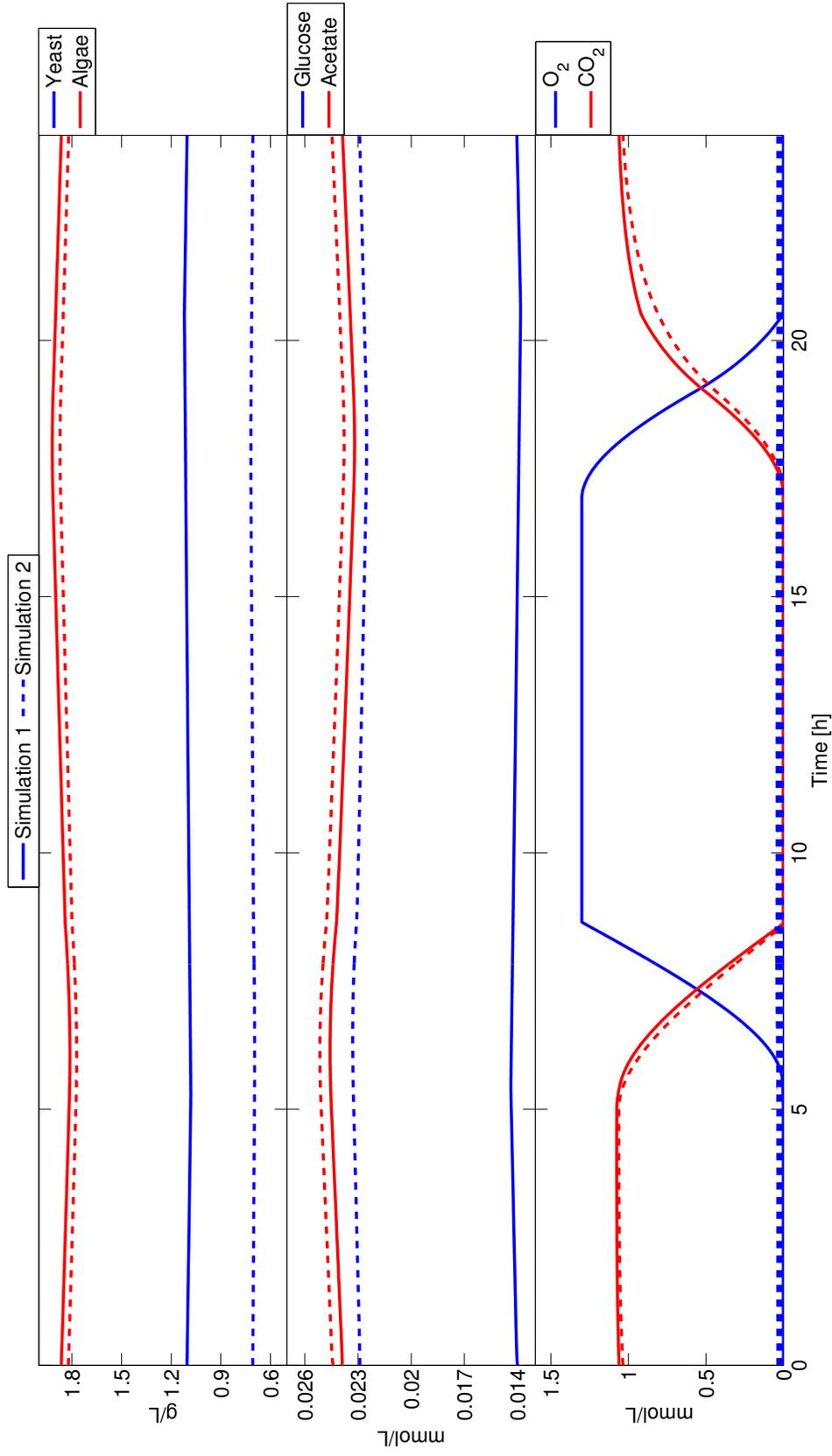
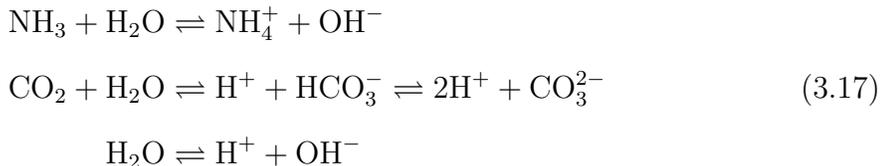


Figure 3-3: DFBAlab simulation results of Example 3.2.2. Two cyclic steady states are presented. Simulation 1 (solid line) was performed with lexicographic objectives presented in Table 3.2, whereas simulation 2 (dashed line) used the negative of Objective 4 for algae. Significant differences can be observed in the predicted concentrations of yeast, glucose, and oxygen. Computation times for simulations 1 and 2 were 16.4 and 15.3 seconds, respectively.

Lexicographic optimization is very important in this example; if the negative of Objective 4 for algae is used, oxygen, acetate and yeast concentration profiles vary significantly. In particular, notice the large difference in the O_2 concentration profile between the two simulations. Since the O_2 flux is nonunique, selecting different fluxes will lead to different trajectories. Without a rule on how to choose a flux from the optimal solution set, DyMMM can choose different elements of this set while cutting its time step, obtaining unreliable right-hand side information. Therefore, it is not surprising that the DyMMM simulator was unable to simulate this system.

It must be noted that in reality, this difference is not observed in nature. When Objective 4 for algae is inverted (maximizing O_2 consumption and minimizing O_2 production), the model is able to uptake unlimited H^+ ions from the environment and produce water until the O_2 uptake bound is reached. This behavior will change the pH of the system and the overconsumption of O_2 would be unsustainable. Increased modeling efforts can bound the uptake of other substrates such as nitrogen, phosphorus and iron and use pH dependent uptakes. In this context, a pH balance will be necessary. This balance is implemented in Example 3.2.3. Finally, biologically relevant lexicographic objectives must be selected because some objectives may lead to unrealistic systems as the one just presented.

Example 3.2.3. This example illustrates the modeling flexibility DFBAlab provides. The growth rate of autotrophic microalgae such as *C. reinhardtii* is dependent on CO_2 concentration. This concentration is affected by pH, as the following equilibrium reactions are present in the extracellular environment:



Using the equilibrium constants presented in Table 1 in [144] and the equilibrium model in Equations (14a) and (14b) in [144], a pH balance was introduced to Example 3.2.2. The pH balance introduces algebraic equations that have to be satisfied at all

times. This kind of system is called a Differential-Algebraic Equation system (DAE) where some variables are algebraic variables (their time derivative is not calculated explicitly) and others are differential variables. To our knowledge, no one has introduced the pH equilibrium equations in a DFBA simulation before. To add the pH balance to the system, total carbon and total nitrogen were added to the differential variables, CO_2 concentration was transformed into an algebraic variable, and new algebraic variables for NH_4^+ , NH_3 , HCO_3^- , CO_3^{2-} , and H^+ concentrations were introduced. Total nitrogen in the system was assumed to be constant at 0.1643 mmol/L, which is the concentration present in the Charles River in Cambridge [20], the effect of H^+ exchange by algae on pH was considered negligible, and ionic valency of the solution was assumed to be equal to zero.

If the Jacobian of the algebraic equations with respect to the algebraic variables is nonsingular, the DAE is index-1 and can be solved with MATLAB ode15s. Table 3.3 shows the initial conditions and parameters used. No non-negativity constraints were enforced; however, the uptake kinetics were specified so that negative concentrations could not occur. Concentration profiles are presented in Figures 3-4 and 3-5. Simulation results with a pH balance are close to those without a pH balance. However, the information obtained from this simulation enables using pH dependent uptake kinetics and ionic species uptake kinetics leading to more accurate simulations. It took only 15.9 seconds to simulate accurately 24 hours of this coculture with a pH balance.

Example 3.2.4. In this example a monoculture of *Chlamydomonas reinhardtii* was simulated to illustrate how DFBAlab performs for simulations with a large number of species models. The parameters implemented in Example 3.2.2 were used with different initial conditions. No non-negativity constraints were enforced, but the uptake kinetics were specified so that negative concentrations could not happen. Algae biomass was split among several LPs and running times were compared. Table 3.4 shows the running times for 24 hours of simulation for different numbers of models in the system.

Table 3.3: Initial concentrations and parameters of Example 3.2.3.

Variable	No pH balance	pH balance		Parameters	
y_0^Y	1.10	1.10	gDW/L	V_0	140 L
y_0^A	1.86	1.87	gDW/L	F_{in}	1 L/h
g_0	1.40 E^{-2}	1.40 E^{-2}	mmol/L	F_{out}	1 L/h
o_0	6.53 E^{-4}	6.52 E^{-4}	mmol/L		
c_0	1.06	1.06	mmol/L		
e_0	8.21	8.21	mmol/L		
a_0	2.39 E^{-2}	2.37 E^{-2}	mmol/L		
N_T	-	1.64 E^{-1}	mmol/L		
C_T	-	1.22	mmol/L		
NH_3	-	2.45 E^{-5}	mmol/L		
NH_4^+	-	1.64 E^{-1}	mmol/L		
HCO_3^-	-	$1.64 \text{ E}^{-1} \text{ E}^{-2}$	mmol/L		
CO_3^{2-}	-	2.58 E^{-6}	mmol/L		
H^+	-	2.67 E^{-6}	mmol/L		

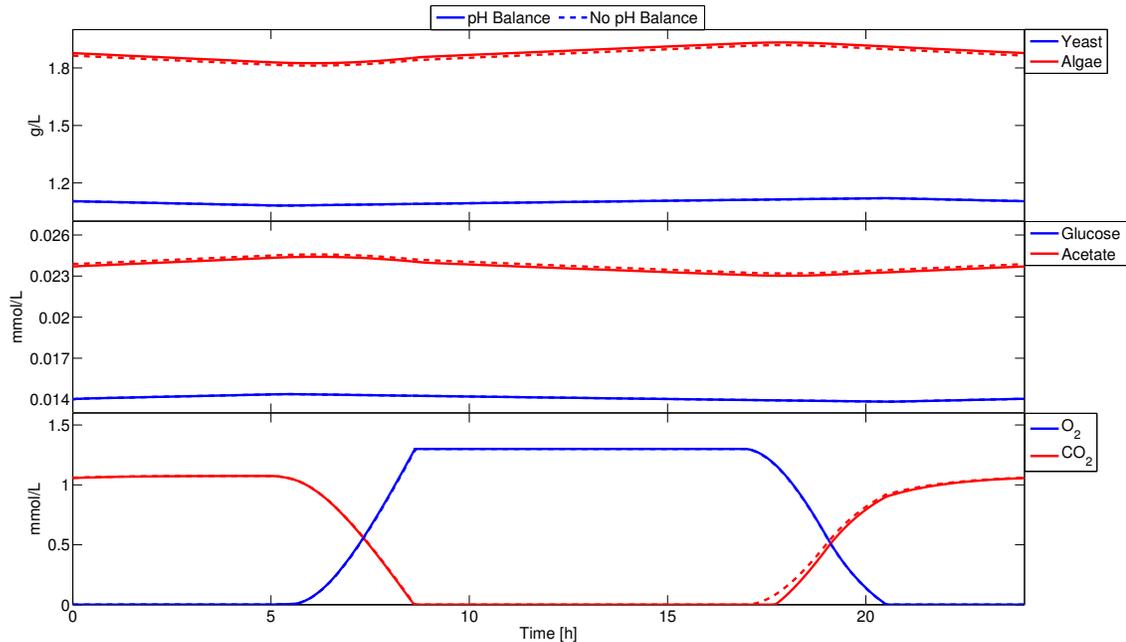


Figure 3-4: DFBAlab simulation results of Example 3.2.3. This example incorporates the pH balance (solid line). Simulation results were close to the ones obtained without a pH balance. Slight variations were observed for the CO_2 concentration profile. Computation time was 15.9 seconds.

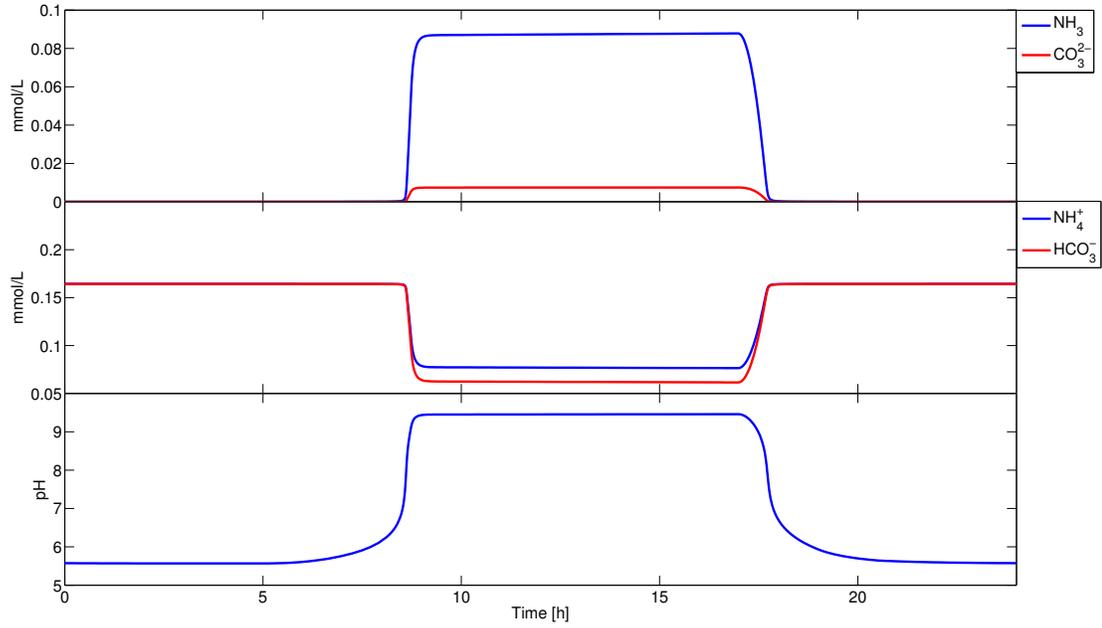


Figure 3-5: Equilibrium species and pH of Example 3.2.3. The pH balance enables tracking of ionic concentration profiles. This information allows using pH dependent uptake kinetics and uptake kinetics for ionic species.

Table 3.4: Running times of Example 3.2.4 with increasing number of models.

Number of models	Time (s)
1	8.1
2	15.6
5	38.2
10	73.7
25	187

3.2.1 Discussion

In these examples, the reliability and speed of DFBA_{lab} has been shown compared to current open MATLAB benchmarks in DFBA simulation. COBRA lacks flexibility when implementing Michaelis-Menten kinetics and the use of a fixed time step decreases the accuracy of these simulations, or increases the integration time for very small time steps. DyMMM provides a flexible framework that allows the implementation of community simulations. However, if any of the exchange fluxes are nonunique, simulation results will be incorrect. DFBA_{lab} uses lexicographic optimization to obtain a well-defined system, but it requires specification of lower-priority objective functions. Biologically relevant lower-priority objectives must be sought to restrict the solution set of (3.2) to a more realistic set. For instance, it has been suggested by a reviewer of [44] that maximization of ATP is a biologically relevant objective that should follow maximization of biomass. In DFBA_{lab}, this objective can be added right after maximization of biomass. Then, the unique exchange fluxes obtained are guaranteed to maximize biomass first, and then maximize ATP. If other biologically relevant objectives are found, they can be added in the same way to the priority list (after maximization of biomass, but before the exchange fluxes), such that the exchange fluxes obtained are more realistic.

The DFBA_{lab} framework is flexible enough to allow DAEs, which could result from performing pH balances in the culture. Furthermore, in community simulations, the running time of DFBA_{lab} increases linearly with the number of LP models when they all correspond to the same species, although it is expected not to follow a linear relationship if a multispecies simulation is carried out because of the interactions between microorganisms. The LP feasibility objective function in DFBA_{lab} serves two purposes: it helps to distinguish between feasible and infeasible trajectories and it can serve as a penalty function in optimization algorithms.

3.3 Conclusions

The objective of this work is to provide an easy to use implementation that minimizes troubleshooting of numerical issues and facilitates focus on the analysis of simulation results. DFBAlab, a reliable DFBA simulator in MATLAB, is presented. DFBAlab uses lexicographic optimization to obtain unique exchange fluxes and a well-defined dynamic system. DFBAlab uses the LP feasibility problem to generate an extended dynamic system and a penalty function. It also uses LP basis information to reformulate the DFBA problem as a sequence in time of DAE systems.

DFBAlab performs better than its counterpart DyMMM in complex community simulations: it is faster and more accurate because the unique fluxes provided by lexicographic optimization are necessary for efficient and reliable numerical integration. In addition, DFBAlab can integrate the DAEs resulting from implementing pH balances. Biologically relevant lower-priority objectives must be sought to perform lexicographic optimization. The penalty function provided by DFBAlab can be used to optimize DFBA systems.

DFBAlab currently has approximately 150 academic users and it has generated some industrial interest. Information on how to use DFBAlab and other relevant documentation can be found in Appendix A.

Chapter 4

Modeling of an algae cultivation system for biofuels production using dynamic flux balance analysis

This chapter is based in the work published in [45] and presented in [46]. This chapter shows how an algal-fungal pond is able to attain higher biomass productivities than the respective monocultures. The substrates required for algae growth are minimal. For algal photoautotrophic growth, CO_2 is the carbon source, energy is provided by sunlight, and small amounts of nitrogen, phosphorus and sulfur sources need to be provided. The quantity of the available substrates strongly determines the growth rate and intracellular accumulation of desired metabolic products such as lipids. For yeast, a carbon source, in this case glucose and xylose, and small amounts of nitrogen, phosphorus and sulfur are required. This case study shows that yeast provides additional CO_2 to algae by metabolizing sugars and algae provides O_2 to yeast. Furthermore, together yeast and algae use available resources more efficiently, which makes the invasion of other microorganisms less likely. This chapter uses the modeling framework presented in Höffner and Barton [58], which is based on dynamic flux balance analysis (DFBA)[137, 98, 87, 100] and the high-rate algal-bacterial pond model [18, 144].

4.1 Methods

The design of novel algal open pond systems requires process models, which provide quantitative predictions of interactions between process components across different scales. Multi-scale models, integrating genome-scale information in metabolic networks with the ecological scale of the interactions between multiple species and the process scale of bioreactors, have been proposed in Höffner and Barton [58]. These complex models are based on multi-species dynamic flux balance analysis and can be used for the discovery of novel and improved microbial bioprocesses.

4.1.1 Dynamic Flux Balance Analysis

Flux balance analysis (FBA) is a genome-scale, constraint-based modeling approach. It is a widely successful framework for metabolic engineering and analysis of metabolic networks [98, 100]. Consequently, metabolic network models of many organisms have been developed [117]. Based on genomic analysis, a metabolism can be modeled as a network of reactions, which must satisfy simple mass balance constraints. The network reconstruction determines the stoichiometry of the metabolism under the balanced growth assumption [100]. However, this network is often underdetermined; the fluxes of the different substrates and metabolites can vary and yet still produce a solution which satisfies mass balance constraints. Thus, it is assumed that the fluxes will be such that some cellular objective is maximized. For example, an evolutionary argument can be made that a microorganism will maximize its growth rate if sufficient nutrients are provided [98].

DFBA combines genome-scale metabolic network analysis with a dynamic simulation of the extracellular environment [137, 87]. At this scale, process models of bioreactors incorporating detailed metabolic network reconstructions can be considered. DFBA models have matched accurately experimental data for the cultivation of *E. coli* [137, 51] and the competition between *Rhodospirillum rubrum* and *Geobacter* [150]. In addition, DFBA has successfully modeled experimentally observed mutualistic relationships between *D. vulgaris* and *M. maripaludis* and between engineered yeast

strains unable to grow on minimal glucose medium separately, and has been used to make fast predictions for combinations of microorganisms and media not yet validated experimentally [77]. DFBA provides a platform for detailed design, control, and optimization of biochemical process technologies, such as an open pond. With DFBA, temporal and/or spatial variations in the behaviour of the community within the bioreactor can be simulated. This formulation provides a more appropriate and predictive description of complex ecological systems, in which emergent nonlinear dynamic behaviour is a common phenomenon. Furthermore, the mathematical formulation allows for unstructured models of ecological species, such as large zooplankton, for which a metabolic model is not available.

Simulation and optimization of large multi-species and multi-scale process models requires efficient numerical tools. A DFBA model results in a dynamic system with linear programs embedded [59, 56]. Numerical complications arise when simulating these systems; these have recently been addressed and efficient simulators have become available [59, 44]. Therefore, simulation of large-scale multi-species metabolic reconstructions is now possible. The simulations in this chapter were performed using DFBAlab [44].

4.1.2 High-Rate Algal Pond Model

The high-rate algal-bacterial pond model was first introduced and validated experimentally by Buhr and Miller [18] and then extended by Yang [144]. This model considers a coculture of bacteria and algae for high-rate wastewater treatment ponds. Their growth expressions are given by Monod type kinetics dependent on the concentration of carbon, oxygen, and nitrogen. In addition, they considered pond depth and biomass concentration effects on light penetration, the effect of ionic species on pH, and the effect of pH on dissolved CO₂. In order to use Monod kinetic expressions, a limiting substrate must be readily identified, and accuracy is lost at transitions, which are characterized by several substrates being limiting. In cocultures and non-steady state environments, predicting limiting substrates and active metabolic pathways can be a very challenging task, if possible. In this chapter we incorporate genome-scale

metabolic models into the high-rate algal-bacterial pond model. When using dynamic flux balance analysis, no a priori predictions are needed because the linear programs modeling the behaviour of each species predict the metabolic state given the extracellular conditions and identify the limiting substrates. Monod kinetics are used indirectly by bounding the consumption of substrates as in Hanly and Henson [51], but the actual consumption rate is calculated by the linear programs after identifying a limiting substrate.

4.1.3 Raceway Open Ponds

A raceway pond is an open pond with flow and can be modelled as a plug flow reactor. In this chapter, the spatial distribution of quantities in the raceway pond is approximated as a sequence of interconnected continuous stirred tank reactors (CSTRs). Each CSTR model includes the mass balances for the main metabolites and an estimate of the variation of the average light intensity during a 24 hour period. For each CSTR, it is assumed that the broth is well mixed such that there are no gradients in nutrients or biomass concentrations. Growth rates of algae and yeast, and uptake and production rates of metabolites are obtained from genome-scale metabolic network reconstructions.

First a pond with an algae monoculture with no CO₂ sparging is analyzed. Next, the productivity of this culture is boosted with CO₂ sparging and a series of three ponds is considered. Next, a pond containing a monoculture of oleaginous yeast is considered and the advantages of an algae/yeast coculture are illustrated. Next, we model an algal/yeast coculture with no flue gas sparging in a three pond system. Finally, the case where the oleaginous yeast can also consume xylose is considered in another three pond system. The coculture examples illustrate the benefits of the symbiotic relationships between yeast and algae. The series of ponds is necessary to induce lipids production through nitrogen starvation [115], as observed experimentally by Rodolfi *et al.* [111] and Breuer *et al.* [16]. Nitrogen starvation increases lipids productivity but reduces biomass productivity [143, 16]. A two phase cultivation system can achieve good biomass and lipids productivity [66]. Therefore, the series of

ponds allows biomass growth in the first pond and lipids accumulation in the latter ones. Ammonia is used as the single nitrogen source. Caustic soda is used to prevent the pond from becoming too acidic.

In this case study, the model for each pond was obtained from Yang [144]. This model considers a 350,000 L outdoor pond with a depth of 0.4 m. It is continuously harvested at a rate of 50,000 L/day with a recycle rate of 350,000 L/h. A channel width of 1.2 m is assumed such that the flow velocity is 0.2 m/s to avoid sedimentation and thermal stratification, as suggested by Becker [10]. The Reynolds number of this pond is of 250,000; turbulent flow is desired to keep cells in suspension and prevent stratification [24]. We discretized the spatial variations of the pond by modeling it as a sequence of nine CSTRs. For ponds connected in series, the effluent of one pond feeds into the next and the effluent of the last pond feeds into a clarifier, in which the water content is reduced and subsequently the remaining biomass is harvested and processed. The clarifier and other downstream processes are not included in the current model.

The average light intensity is estimated based on the Beer-Lambert law [18, 144]:

$$I_a(t) = \frac{1}{L} \int_0^L I_0(t) \exp(-K_e(X(t))z) dz, \quad (4.1)$$

where $K_e(X(t))$ is the extinction coefficient, L is the depth of the pond, and I_0 is the surface light intensity during the photoperiod (7:00-19:00) approximated by a sinusoidal function with maximal intensity at noon and average surface light intensity of 18.81 MJ/m²/day or 5.22 kWh/m²/day[144]. This solar intensity can be found in southwestern USA (see Figure 4-1). To convert to mmol photons/gDW/h, the average cell diameter used was 10 μ m [52], and the average weight was estimated as 10⁹ cells in one gram dry weight [54]. Following the calculations in Boelee *et al.* [13], $I_0^{\max} = 283$ mmol/gDW/h. The dependency of K_e on biomass concentration is modeled via a simple linear relationship,

$$K_e(X(t)) = K_{e1} + K_{e2}X(t), \quad (4.2)$$

where $X(t)$ is the total biomass concentration at time t and the values of the parameters K_{e1} and K_{e2} are taken from Buhr and Miller [18]. In addition, light available for photosynthesis cannot exceed the average surface light intensity of 3610 mmol photons/(m²×h). Therefore,

$$\begin{aligned}
 I_0(t) &= \max \left(0, 283\pi \left(\sin \left(\frac{\pi(t-7)}{12} \right) \right) \right), \\
 I_a(t) &= I_0(t) \times \frac{1 - e^{-LK_e(X(t))}}{LK_e(X(t))}, \\
 I_1(t) &= \frac{\max \left(0, 3610\pi \left(\sin \left(\frac{\pi(t-7)}{12} \right) \right) \right)}{400X_A(t)}, \\
 I_m(t) &= \min(I_a(t), I_1(t)),
 \end{aligned} \tag{4.3}$$

where $I_m(t)$ is the light available for algae at time t in mmol/gDW/h, $X_A(t)$ is algal biomass concentration in g/L and 400 is a conversion factor from g/L to g/m² based on the geometry of this pond. The open pond is in direct contact with the atmosphere, therefore a simple model based on film theory is used to estimate the mass transfer across the interface between air and water with parameters from Buhr and Miller [18] and Yang [144], and pond mass transfer area to volume ratio of 2.5 m²/m³. The equilibrium concentrations for both O₂ and CO₂ in water are calculated using Henry's law. Finally, the dissolved gas concentrations are limited by their saturation concentration at ambient conditions.

Sparging of flue gas is modeled according to Yang [144]. The model considers that flue gas is fed at atmospheric pressure into orifices with a diameter of 5 cm. covering the entire bottom of the pond with a concentration of 250 orifices/m². Flue gas flowrates of 10, 40, 100, 500, and 2000 m³/h were modeled. The flue gas composition of 13.6% CO₂, 5% O₂, and the rest N₂ was obtained from Brown [17]. Variations of the concentration of CO₂ in the gas bubbles with respect to pond depth were considered.

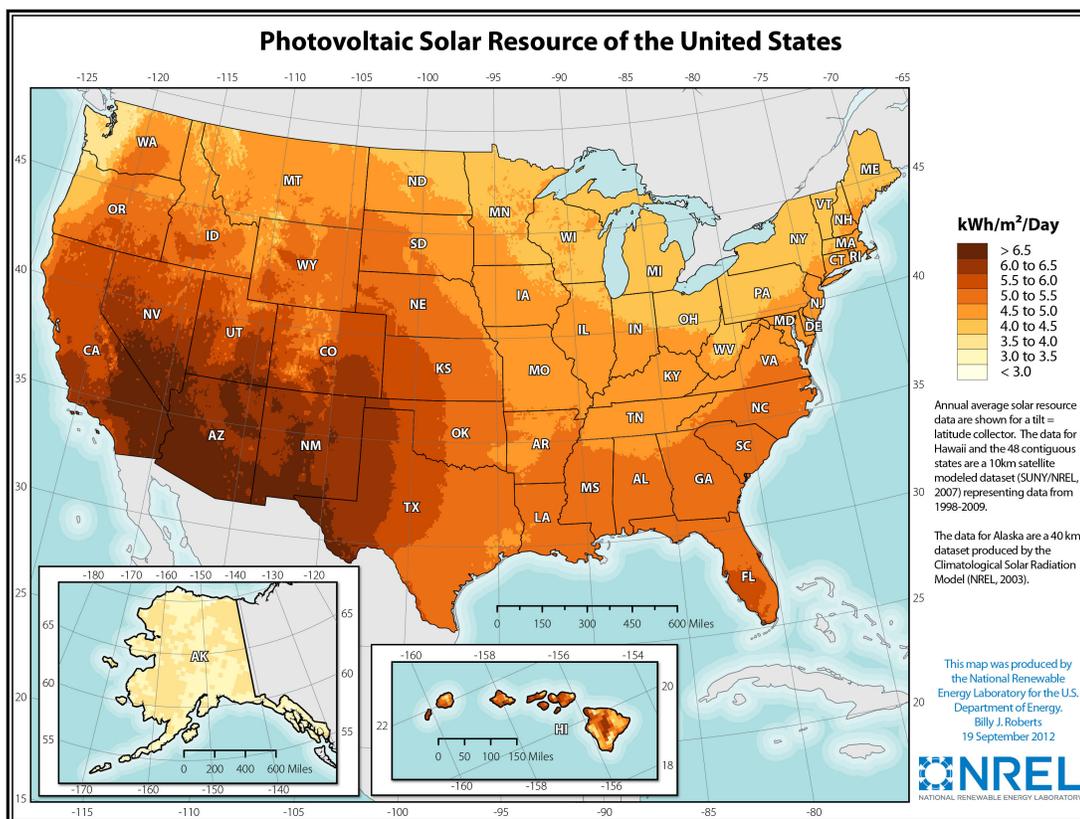


Figure 4-1: Photovoltaic Solar Resource of the United States. This map shows the average surface light intensity in different parts of the United States. Notice that in this work a light intensity of 5.2 kWh/m²/day is assumed. This average light intensity and higher light intensities can be found in southwestern United States. Figure obtained from [95].

4.1.4 Metabolic Models

Chlamydomonas reinhardtii is used as a model organism for microalgae. The genome-scale metabolic network iRC1080 is an up-to-date metabolic reconstruction of *C. reinhardtii* [19]. The reconstruction consists of 2190 fluxes and 1068 unique metabolites, and encompasses ten compartments including a detailed reconstruction of the lipid metabolism. The model includes photoautotrophic, heterotrophic and mixotrophic growth options and a detailed model of the light spectrum. The model predictions have been validated experimentally under different environmental conditions, such as

nitrogen-limited or light-limited growth [19]. The model includes the pathways necessary for the biosynthesis of unsaturated fatty acids, fatty acids, steroids, sphingolipids, glycerophospholipids, and glycerolipids, and it considers the pathways related to fatty acid elongation in the mitochondria. The model considers all individual metabolites in these pathways including backbone molecules, stereochemical numbering of acyl-chain positions, acyl-chain length, and *cis-trans* stereoisomerisms [19]. More model details including a list of all metabolites and reactions can be found in the Supplementary Information of Chang *et al.* [19] and more information in general on algal lipids synthesis in Harwood and Guschina [55]. For this chapter, 125 metabolites were classified as lipids and a dynamic lipid storage was implemented in the model. In addition, minor modifications were done to the metabolic network reconstruction to satisfy mass balances.

The model for the yeast organism is based on a well-established model of *Saccharomyces cerevisiae*. The genome-scale network reconstruction of the *S. cerevisiae* metabolism iND750 has shown good agreement with experimental data [31]. It considers 1061 unique metabolites in eight compartments and 1266 intracellular and exchange fluxes. Furthermore, the model correctly predicts ethanol production under anaerobic conditions. However, *S. cerevisiae* is not an oleaginous yeast. Examples of oleaginous yeasts include *Cryptococcus albidus*, *Lipomyces starkeyi*, *Rhodotorula glutinis*, *Trichosporon pullulans*, and *Yarrowia Lipolytica* with lipid accumulations ranging from 36% to 72% [105, 11]. A description of the lipids profiles for different fungal species can be found in Ratledge [105]. The iND750 model considers most pathways found in fungal species. It also considers the production of different lipids species such as glycerolipids, glycolipids, sphingolipids, phospholipids, and fatty acids. This metabolic reconstruction can be used to model different species by adjusting the biomass equation and adjusting the flux bounds on reactions feeding to different pathways. In this chapter, we modified the iND750 model such that it cannot produce ethanol [11] and under low oxygen conditions it can produce acetate, formate, succinate, and citrate, reflecting the behavior of *Y. Lipolytica* [101]. We also modified it further such that it consumed xylose reflecting the behavior of *Rhodotorula glutinis*

[146]. Therefore, the biomass equation was modified such that the yeast accumulates 40% lipids.

Both modified models are provided as supplementary materials in [45]. Figure 4-2 presents a simplified version of both models. Yeast consumes glucose, xylose, O₂, and nutrients to obtain biomass, CO₂ and water. Under low O₂ conditions, yeast produces acetate (not ethanol because *Y. lipolytica* and *R. glutinis* do not produce ethanol). The metabolic reactions of glucose and xylose generate ATP with stoichiometry defined by the metabolic model. Xylose, glucose and nutrients are assimilated into biomass; these growth reactions have ATP requirements with coefficients determined by the metabolic model. Meanwhile, algae obtains ATP from light and converts CO₂ and water into glucose and O₂ through photosynthesis with some ATP requirement. This glucose can be transformed into starch for energy storage, consumed for ATP production, or assimilated with nutrients as biomass. Under nitrogen limitations, this glucose can be assimilated as lipids. Also, algae can grow heterotrophically on acetate. In addition, the algae model considers a survival ATP requirement. The red, purple, and green arrows show symbiotic opportunities. All ATP coefficients are determined by the metabolic model. Both models, iRC1080 and iND750, contain in full detail all the relevant metabolic pathways that achieve these main reactions. The full list of metabolites and reactions of iRC1080 and iND750 can be found in the Supplementary Information of Chang *et al.* [19] and the Supplemental Material of Duarte *et al.* [31], respectively.

4.1.5 Kinetic Parameters

The uptake kinetics for the exchange fluxes for both microorganisms are approximated by Michaelis-Menten kinetics:

$$v_S = \left(\frac{v_{\max} S}{K_m + S} \right) \left(\frac{1}{1 + I/K_I} \right) (\beta), \quad (4.4)$$

where S is a substrate of interest, I is an inhibitor and β is a positive pH factor. The values of these constants are taken from the literature and presented in Tables

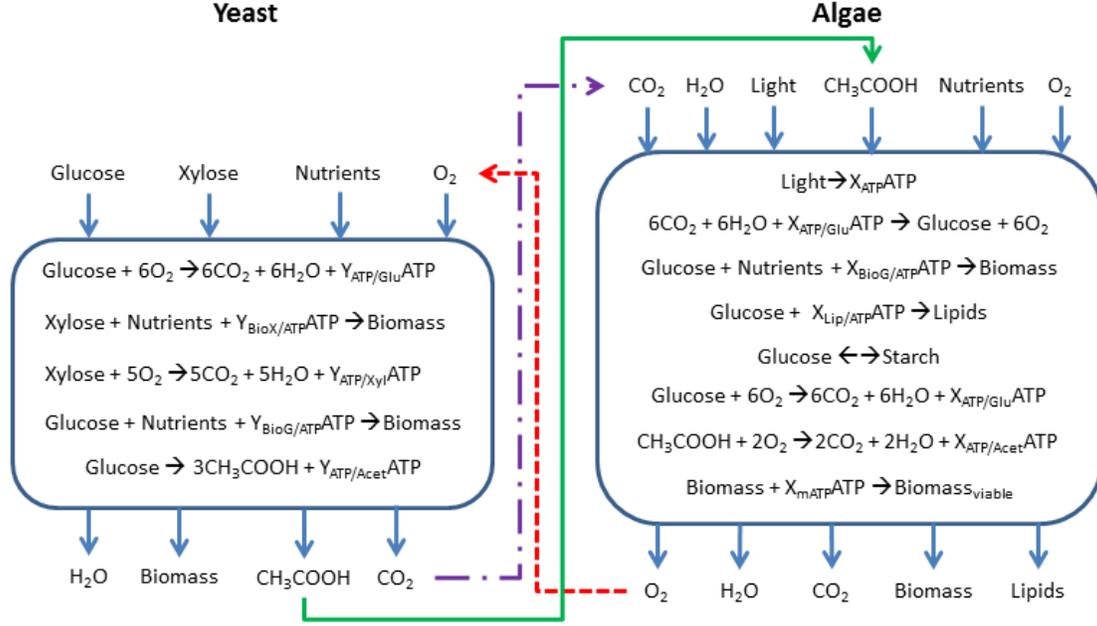


Figure 4-2: Global reactions considered in the modified models iRC1080 and iND750. The stoichiometric coefficients of ATP production and consumption are determined by the metabolic models. The colored arrows illustrate symbiotic relationships: the red dashed arrow shows yeast utilizing O₂ produced by algae, the purple dotted and dashed arrow shows algae consuming CO₂ produced by yeast, and the solid green arrow shows that the acetate produced by yeast can be metabolized by algae.

4.1 and 4.2. The uptake of acetate in algae was modeled according to Zhang using the expression for growth on ammonium chloride [147], but since this expression is slightly different from (4.4), the values of its constants are not reported in Table 4.1. Algae are known to survive in a pH range of 6-10 with an optimum pH of 8, whereas yeasts survive in environments with pH ranging from 2 to 8 [140]. Algal carbon and nitrogen uptakes and yeast glucose and xylose uptakes were made pH dependent with expressions obtained from Tang *et al.* [127] and Zhang *et al.* [147]:

$$\beta = \left(\frac{\alpha}{K_1 + \frac{K_{OH^-}}{H^+} + \frac{H^+}{K_{H^+}}} \right), \quad (4.5)$$

where α , K_1 , K_{OH^-} , and K_{H^+} are constants. Algal pH dependent growth data under nitrogen and carbon limitations was obtained from Franco *et al.* [41] and Kong *et al.* [79], whereas yeast parameters were adjusted such that it grew at pH levels between 2

Table 4.1: Summary of uptake kinetic parameters for algae and yeast.

Yeast	v_{\max} ($\frac{\text{mmol}}{\text{gDW} \times \text{h}}$)	K_m (mmol/L)	K_I (mmol/L)	Ref.
Glucose	22.4	4.44	EtOH: 217	Hanly <i>et al.</i> [51]
O ₂	2.5	0.003	None	Hanly <i>et al.</i> [51]
NH ₄ ⁺	25.5	35.4 × 10 ⁻³	None	Jongbloed <i>et al.</i> [68]
Xylose	12.8	32.5	EtOH:217 Glucose:2.78	Hanly <i>et al.</i> [51]
Algae				
CO ₂	1.25	0.03	None	Tsuzuki <i>et al.</i> [131]
O ₂	2.065	0.008	None	Yang [144]
HCO ₃ ⁻	1.82	0.27	None	Tsuzuki <i>et al.</i> [131]
NH ₄ ⁺	0.65	3.84 × 10 ⁻⁴	None	Hein <i>et al.</i> [57]
NO ₃ ⁻	0.251	1.1 × 10 ⁻³	None	Galván <i>et al.</i> [42]
Acetate	N.A	N.A	None	Zhang <i>et al.</i> [147]

Note: Ammonium uptake for yeast was approximated with that of fungus *Lactarius rufus*. The weight fraction of chlorophyll (22.8 mg/gDW) in algae was obtained from the biomass equation in the iRC1080 model.

The yeast uptake of xylose was scaled from E.coli values.

and 9 with maximum growth rate at pH equal to 6. Table 4.2 presents the constants used for these simulations.

Table 4.2: Constants for pH dependent uptakes of algae.

Algae	NH ₄ ⁺ , NO ₃ ⁻	CO ₂ , HCO ₃ ⁻	Acetate
α	1.25	1.08	1
K_1	1.19	1.07	1.18
K_{OH^-} [mol/L]	3.51 × 10 ⁻¹⁰	9.26 × 10 ⁻¹¹	2.82 × 10 ⁻⁸
K_{H^+} [mol/L]	3.19 × 10 ⁻⁷	5.9 × 10 ⁻⁶	6.66 × 10 ⁻⁸
Yeast	Glucose/Xylose		
α	1		
K_1	0.97		
K_{OH^-} [mol/L]	2 × 10 ⁻⁸		
K_{H^+} [mol/L]	1 × 10 ⁻⁴		

Finally, an expression was derived for algal starch production and consumption. In some simulations, starch is the only source of energy for algae at night. It is assumed that starch production is dependent on light and carbon concentrations and

that its consumption is dependent on intracellular starch concentration at night:

$$v_{starch}^{production} = \begin{cases} 9.5 \times 10^{-4}(\gamma) + Y_{starch}\mu & \text{when } I_0(t) \geq 300, \\ 9.5 \times 10^{-4}(\gamma) \left(\frac{I_0}{300}\right) + Y_{starch}\mu & \text{otherwise,} \end{cases} \quad (4.6)$$

$$\text{where } \gamma = \min \left(0.1, \left(\frac{[\text{HCO}_3^-]}{K_m^{\text{HCO}_3^-} + [\text{HCO}_3^-]} + \frac{[\text{CO}_2]}{K_m^{\text{CO}_2} + [\text{CO}_2]} \right) \beta_{\text{CO}_2} \right),$$

$$v_{starch}^{consumption} = \begin{cases} 0 & \text{when } I_0(t) \geq 300, \\ \frac{1.4 \times 10^{-4} S}{S + 0.006} \left(1 - \frac{I_0}{300}\right) & \text{otherwise,} \end{cases} \quad (4.7)$$

where $v_{starch}^{production}$ and $v_{starch}^{consumption}$ are given in mmol/gDW/h, $[\text{HCO}_3^-]$ and $[\text{CO}_2]$ are concentrations in mmol/L, μ is the growth rate in mmol/gDW/h, Y_{starch} is the starch fraction of the biomass growth equation, S is the intracellular starch concentration in mmol/gDW, and β_{CO_2} is the pH factor for CO_2 uptake in algae. The constants in Equations (4.6) and (4.7) were fine-tuned by running several simulations. Algae growth rate depends on light intensity, dissolved O_2 , CO_2 , HCO_3^- , acetate, and NH_4^+ concentrations. Yeast growth rate depends on dissolved O_2 , glucose, and NH_4^+ concentrations. Both growth rates are determined through FBA.

4.1.6 Solution Equilibrium

The chemical equilibrium of the system is based on Buhr and Miller [18] with parameters obtained from Robinson and Stokes [109] at 20 °C. It is assumed that the ions present in the system are CO_3^{2-} , H^+ , OH^- , HCO_3^- , NH_4^+ , Na^+ , and the ions resulting from formic, acetic, succinic, and citric acids. Ammonia is assumed to dissolve completely. Therefore, a system of equations is obtained from the solution equilibria of ammonia, carbon dioxide, formic, acetic, succinic, and citric acids and water, the mass balances of ammonia, carbon, acetate, formate, succinate, and citrate, and electroneutrality. From this system of equations, the concentrations of all ionic species are obtained.

4.1.7 DFBAlab Hierarchy of Objectives

Since the flux distribution associated with maximal growth is not necessarily unique, hierarchical optimization is used to determine unique exchange fluxes among the optimal flux distributions. DFBAlab requires the user to provide a hierarchy of objectives for efficient integration of the dynamic system [44]. Table 4.3 presents the objectives used for algae and yeast.

Table 4.3: Hierarchy of objectives for simulation with DFBAlab.

	Yeast	Algae
1	Maximize growth	Maximize autotrophic growth
2	Maximize CO ₂ production	Maximize lipids production
3	Maximize glucose consumption	Maximize starch production
4	Maximize xylose consumption	Maximize consumption and minimize production of CO ₂
5	Maximize O ₂ consumption	Maximize ammonium consumption
6	Maximize ammonium consumption	Maximize HCO ₃ ⁻ consumption
7	Minimize acetate production	Maximize consumption and minimize production of O ₂
8	Minimize formate production	Minimize formate production
9	Minimize citrate production	Minimize ethanol production
10	Minimize succinate production	Minimize acetate production
11		Minimize hydrogen production

4.2 Results and Discussion

In this section, some quantities are reported per m² of illuminated area. This is a common normalization quantity that allows performance comparison with algal production processes reported in the literature.

4.2.1 Algae monoculture without CO₂ sparging

First an algae monoculture with no CO₂ sparging is simulated. It is supplemented with 146 mg/(m² × day) of ammonia. This amount of nitrogen is enough for the pond to be carbon-limited. Figure 4-3 presents a schematic of the simulation. The 350,000 L raceway pond is approximated by nine CSTRs of equal volume.

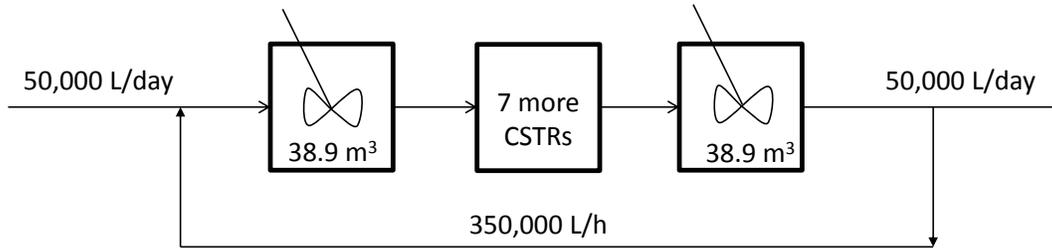


Figure 4-3: Schematic of the raceway pond model. The 350 m³ pond is approximated by nine CSTRs. There is a constant feed and outlet of 50 m³/day and a recirculation of 350 m³/h.

The results of this simulation show that all sections of the pond have very similar concentration profiles. This is a consequence of having a recycle rate 168 times greater than the dilution rate. In fact, a plug flow reactor with a very high recycle rate can be approximated by a single CSTR. Therefore, we modeled the pond as a single CSTR and compared the results with the approximation of 9 CSTRs. The predicted outflow concentration profiles of both approximations are very similar. Therefore, all ponds in the following case studies are modeled as single CSTRs.

Figure 4-4 shows the predicted concentration profiles in the pond at cyclic steady state. It can be seen that the predicted biomass productivity is less than 1 g/(m² × day). The cyclic nature of the steady state can be observed in the concentration profiles of O₂ and CO₂ as well as in the pH of the pond. During the day, algae produces O₂ and consumes CO₂ which increases the pH due to the depletion of carbonic acid; the opposite behavior takes place at night. Due to the low predicted productivity of an algae monoculture without additional CO₂ supply, this system is not explored any further. The next case study is that of an algae monoculture with sparging of flue gas.

4.2.2 Algae monoculture with CO₂ sparging

A schematic of the cultivation system can be observed in Figure 4-5. With flue gas sparging (13.6% CO₂, 5% O₂, and the rest N₂), biomass productivity increases greatly as more CO₂ is supplied into the system. Flue gas was fed into a three pond system

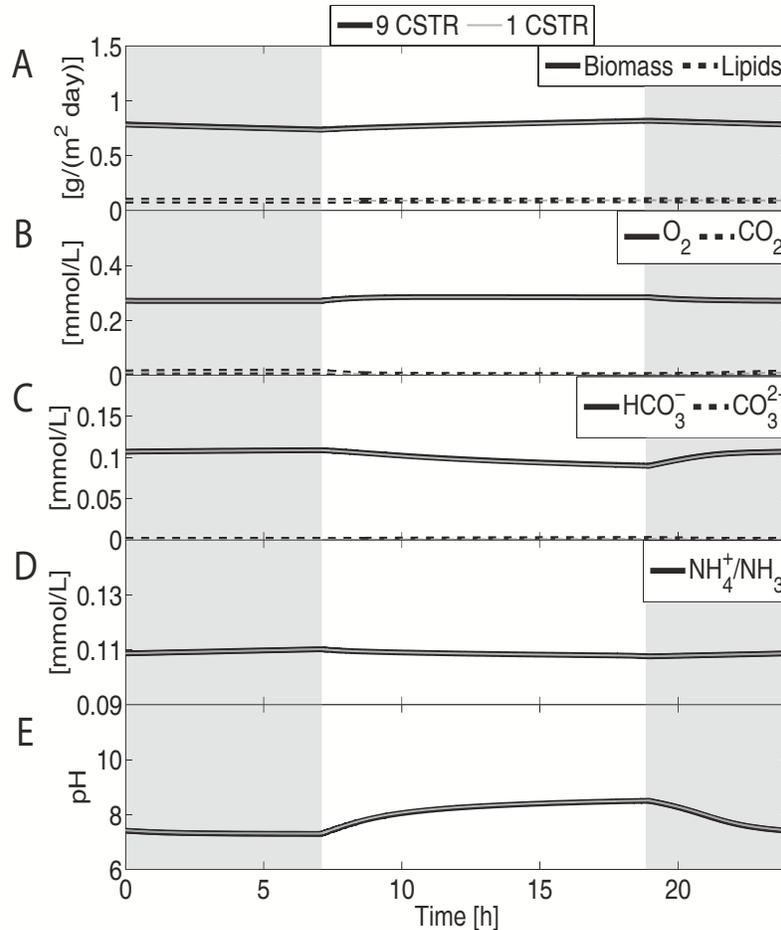


Figure 4-4: Concentration profiles of an algae monoculture pond with no CO_2 sparging. Shaded areas represent dark periods. Notice that the results are the same for a simulation discretizing the length dimension of the pond as 9 CSTRs and one modeling the pond as a single CSTR. A) Predicted biomass and lipids productivity is approximately 0.78 and 0.09 $\text{g}/(\text{m}^2 \times \text{day})$, respectively. B) The photosynthetic activity of algae slightly increases O_2 and reduces CO_2 concentrations during the day. The opposite behavior occurs at night. C) A small amount of HCO_3^- is metabolized by the monoculture. D) Nitrogen sources are consumed faster during growth periods, causing their concentrations to drop during the day. The pond is not nitrogen-limited. E) pH increases during the day as the concentration of CO_2 drops and decreases at night as CO_2 is accumulated again. The pH stays between 7 and 9.

for 10 hours during the day at a sparging rate of $40 \text{ m}^3/\text{h}$. A total of 1.04 and 0.15 $\text{g}/(\text{m}^2 \times \text{day})$ of ammonia and sodium hydroxide, respectively, are fed into the system. Figure 4-5 shows how these feeds are distributed among the three ponds. With this feed distribution, the last pond is nitrogen-limited, and lipids production is induced.

Figure 4-6 shows that this cultivation scheme can attain biomass and lipids pro-

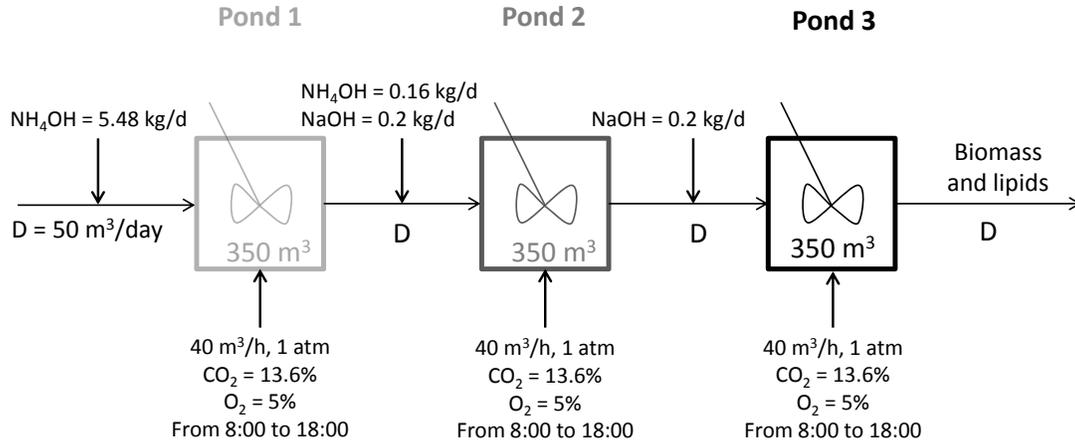


Figure 4-5: Schematic of the algal biomass cultivation system using three raceway ponds. Each pond can be modeled as a CSTR with a volume of 350 m³. There is a constant feed and outlet of 50 m³/day for each pond. The last two ponds present nitrogen limitations inducing lipids production. Sodium hydroxide is fed at a constant rate all day long, whereas ammonia is fed from 8:00 to 18:00. Flue gas sparging occurs only from 8:00 to 18:00.

ductivities of 34.4 and 16.2 g/(m² × day), respectively, which is in line with the 20-40 g/(m² × day) observed in several raceway ponds in the last decade as reported by Figure 20 in Williams and Laurens [143]. The level of accumulation of biomass is highly dependent on the feed rate of flue gas, for low feed rates, and the concentration of CO₂ in this gas. The accumulation of biomass is most likely an upper bound on what can be obtained realistically as the effects of invading species or of toxic components in low concentrations in the flue gas have not been included.

Figure 4-7 shows how biomass and lipids concentrations increase at each pond. Due to nitrogen starvation, the last pond accumulates a higher weight fraction of lipids. In addition, biomass accumulation is slower as the lipid fraction increases, as reported by Williams and Laurens [143]. The carbon atom balance is presented in Table 4.4 and in Figure 4-8; most of the carbon in the flue gas is fixed into algal biomass.

Increasing the flue gas flowrate increases biomass concentration until the culture becomes light-limited. Considering a theoretical limit in sunlight capture by algae of 10%, [143] an average sunlight energy of 6.3 × 10⁶ kJ/(m² × year) [3], and an average algal biomass calorific value of 24.7 kJ/g [143], the maximum possible yield of algae

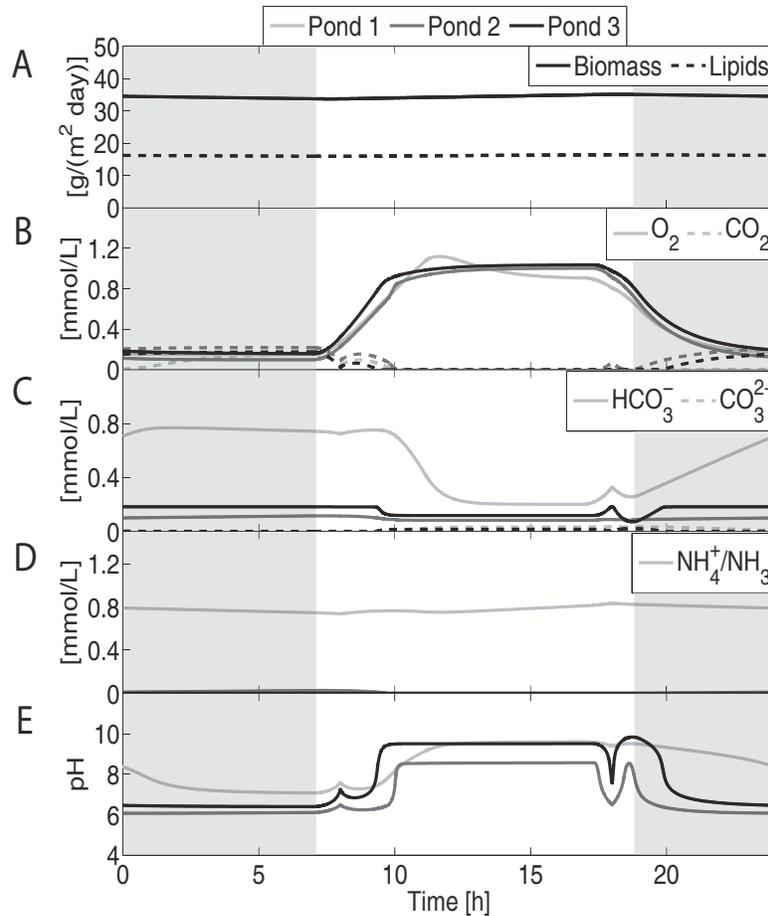


Figure 4-6: Concentration profiles of an algae cultivation system using three raceway ponds with flue gas sparging. Shaded areas represent dark periods. A) Predicted biomass and lipids productivities are approximately 34.4 and 16.2 g/(m² × day), respectively. B) Due to algae’s photosynthetic activity, O₂ concentration increases during the day and decreases during the night, whereas CO₂ concentration decreases during the day and increases during the night. C) HCO₃⁻ concentration is highly related to pH. D) The concentration of nitrogen drops as we move from Pond 1 to Pond 3. Pond 3 is effectively nitrogen-limited inducing lipids production. E) The pH of the system ranges from 6 to 10. For Pond 1, pH is mostly influenced by the concentration of NH₄⁺, whereas for Ponds 2 and 3, it is mostly influenced by the concentration of CO₂.

would be approximately 70 g/(m² × day). Table 4.4 shows the results when the feed rate is increased from 40 to 100, 500, and 2000 m³/h, respectively. The maximum biomass productivity predicted by the model is about 52 g/(m² × day). As the flue gas feed rate is increased, more carbon is lost to the atmosphere. Therefore, for this to be a viable carbon capture alternative, sparging rates should not be increased beyond

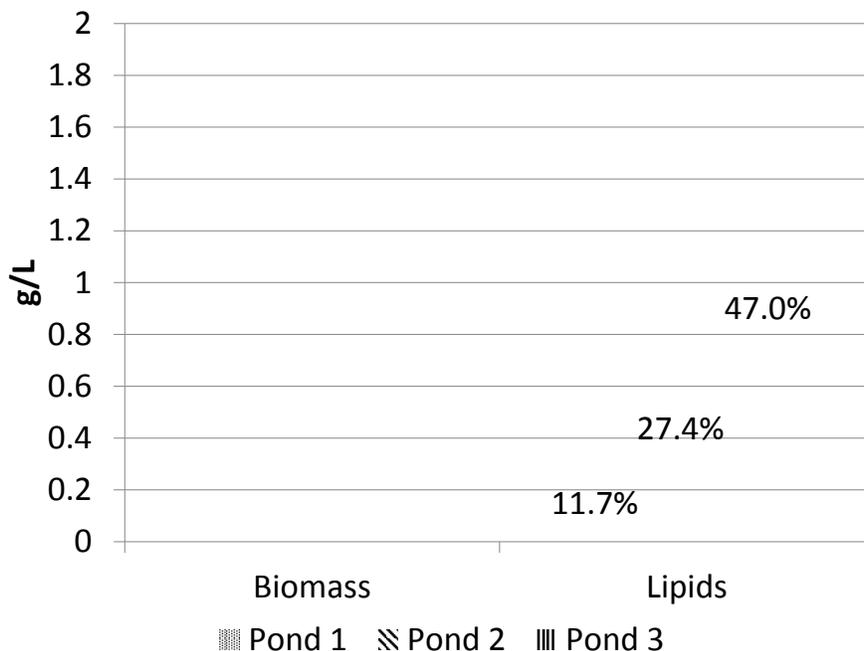


Figure 4-7: Biomass and lipids concentrations in an algae cultivation system using three raceway ponds with flue gas sparging. The numbers on top of the Lipids bars represent the weight fraction of lipids in algal biomass. As a consequence of nitrogen limitations in the last two ponds, the model predicts significant lipids accumulation (up to $\approx 47\%$ weight).

Table 4.4: Carbon balance of an algal monoculture with flue gas sparging

Flue gas feed rate (m^3/h)	10	40	100	500	2000
Algal biomass	76.9%	56.5%	32.1%	6.9%	1.7%
Not transferred from flue gas	16.3%	31.2%	54.0%	83.8%	93.9%
Net loss to atmosphere	6.3%	12.2%	13.8%	9.2%	4.4%
Dissolved inorganic carbon lost in outlet flow	0.5%	0.2%	0.1%	0.1%	$\approx 0\%$
Formate production	$\approx 0\%$				
Biomass $\text{g}/(\text{m}^2 \times \text{day})$	11.7	34.4	48.8	52.1	52.1
Lipids $\text{g}/(\text{m}^2 \times \text{day})$	5.5	16.2	22.9	25.0	24.9

the point where more than half of the carbon is lost to the atmosphere.

This kind of cultivation system can only be considered in locations able to supply considerable amounts of flue gas, for example, near power plants. Otherwise, algae growth is carbon-limited and very low productivities can be predicted as illustrated by the example in Section 4.2.1. A different strategy to deal with this carbon limitation

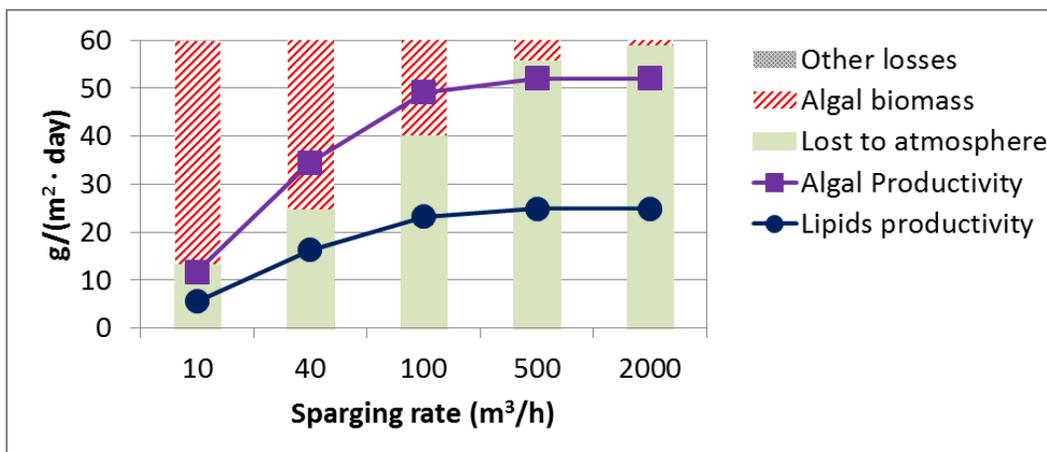


Figure 4-8: Algal biomass/lipids productivity and carbon balance for different sparging rates.

is to feed cellulosic glucose and/or xylose into a pond cultivating an oleaginous yeast. Next, we explore the productivity of an oleaginous yeast monoculture growing on glucose and we illustrate the advantages of having a yeast/algae coculture.

4.2.3 Oleaginous yeast growing on glucose

Oleaginous yeasts, such as *Yarrowia Lipolytica* or *Rhodotorula Glutinis*, can be cultivated as monocultures growing on glucose [115]. However, the maximum concentration of biomass is limited by the mass transfer rate of O_2 from the atmosphere. For an open pond with the same characteristics as the previous examples, DFBAlab predicts a maximum concentration of yeast of 1.82 g/L. However, significant amounts of acetate are produced. The presence of acetate indicates a shortage of O_2 in the system. The maximum biomass concentration attained with no production of acetate is 1.57 g/L. Given the higher efficiency on the utilization of glucose when the yeast is cultivated with sufficient O_2 , the inputs the following simulations were tailored such that no acetate was produced.

As discussed earlier and illustrated in Figure 4-2, the O_2 limitation in yeast and the CO_2 limitation in algae create a great opportunity. If both species are cultivated together, the yeast can benefit from the O_2 produced by the algae and the algae can benefit from the CO_2 produced by the yeast. As a proof of concept, let us assume

that both species are cultivated together with a constant source of sunlight of 3.6 moles of photons/(m² × h) corresponding to 6.8 × 10⁶ kJ/(m² × year) [144]. A total biomass loading of 2.54 g/L can be attained. A summary of these cases is presented in Table 4.5. It is clear that the coculture presents advantages in resource utilization and total biomass productivity with respect to the monocultures. Next, we explore the behavior of an algal/yeast coculture feeding pure glucose and a mix of glucose and xylose into a raceway pond.

Table 4.5: Inputs and outputs for yeast monocultures and yeast/algal cocultures with constant light.

	Case 1	Case 2	Case 3
Inputs [g/(m² × day)]			
Glucose	200.6	300.9	195.4
NH ₃	11.4	17.1	17.4
NaOH	0	34.3	0
Outputs [g/(m² × day)]			
Yeast	89.6	104.1	87.3
Algae	0	0	57.7
Total biomass	89.6	104.1	145
Acetate	0	73.3	0
Carbon Balance Outputs			
Yeast	62.8%	48.6%	59.6%
Algae	0%	0%	35.6%
Acetate	0%	24.8%	0%
CO ₂ lost to atmosphere	36.7%	26.0%	4.6%
Lost in flow	0.5%	0.6%	0.2%

4.2.4 Algae/yeast coculture with cellulosic glucose feed

Figure 4-9 presents a schematic of the cultivation system. Cellulosic glucose is metabolized by yeast and converted to CO₂ which is then fixed by algae. A total of 94.4, 3.1, and 0.03 g/(m² × day) of glucose, ammonia, and sodium hydroxide, respectively, are fed into the system. Figure 4-9 shows how these feeds are distributed among the three ponds; the last pond is nitrogen-limited, and lipids production is induced.

Figure 4-10 shows that this cultivation scheme can attain yeast, algae, and lipids productivities of 34.5, 26.2, and 22.6 g/(m² × day), respectively. Yeast and algae accumulate lipids up to approximately 40% and 33% dry weight, respectively. In the

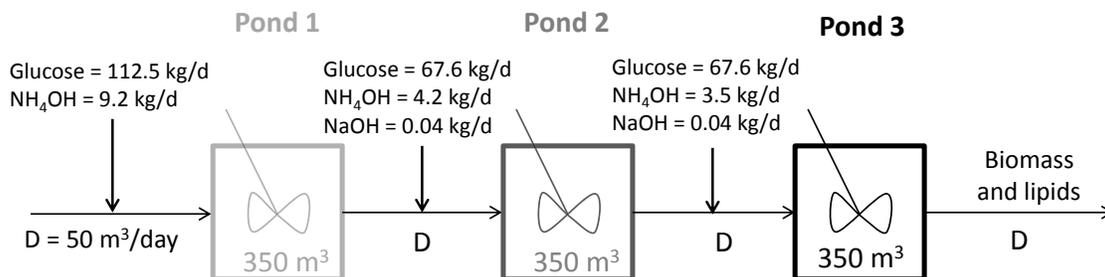


Figure 4-9: Schematic of the algae/yeast cultivation system using cellulosic glucose and three raceway ponds. Each pond can be modeled as a CSTR with a volume of 350 m³. There is a constant feed and outlet of 50 m³/day for each pond. Sodium hydroxide is fed at a constant rate all 24 hours a day, whereas glucose and ammonia are fed only during daytime (12 hours).

coculture case resources are better utilized, making invasion by foreign species more difficult [70]. Figure 4-11 shows algae, yeast and lipids concentrations in each pond. Due to nitrogen limitations in the last two ponds, lipids are accumulated. Table 4.6 presents the carbon balance for this case; approximately 84.7% of carbon in glucose ends in biomass.

Table 4.6: Carbon balance of coculture with pure glucose feed

Carbon Inputs		Carbon Outputs	
Glucose	100%	Yeast biomass	51.4%
		Algal biomass	33.3%
		Net loss to atmosphere	15.2%
		Inorganic carbon lost in flow	0.08%
		Glucose lost in flow	0.02%

4.2.5 Algae/yeast coculture with cellulosic glucose and xylose feed and no acetate production

When cellulosic biomass is hydrolyzed, both glucose and xylose are obtained. Their ratio is dependent on the source of the lignocellulosic waste. A 2 to 1 glucose to xylose ratio by weight is typical [51]. A process that can utilize both, glucose and xylose, is desirable because the sugar mix is cheaper than pure glucose. Some oleaginous yeasts, for example, *Rhodotorula glutinis*, are able to metabolize xylose [146]. Therefore, we

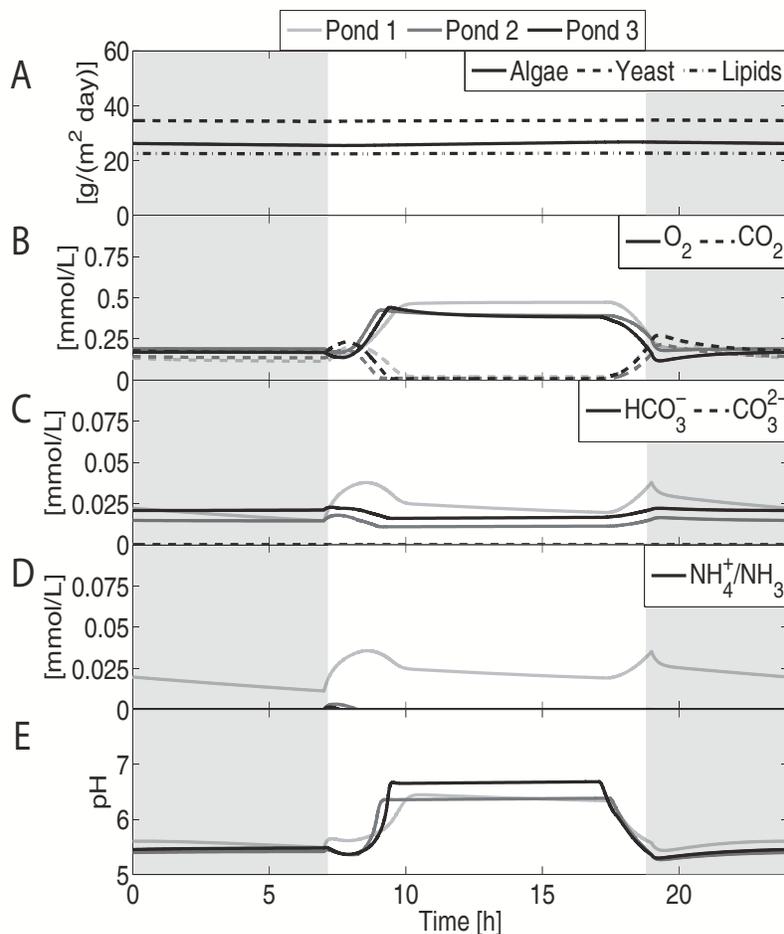


Figure 4-10: Concentration profiles of an algae/yeast cultivation system using three raceway ponds with cellulosic glucose. Shaded areas represent dark periods. A) Predicted yeast, algae, and lipids productivities are approximately 34.5, 26.2, and 22.6 g/(m² × day), respectively. B) Due to algae's photosynthetic activity, O₂ concentration increases during the day and decreases during the night, whereas CO₂ concentration decreases during the day and increases during the night. C) HCO₃⁻ and CO₃²⁻ concentrations remain low. D) The concentration of nitrogen drops as we move from Pond 1 to Pond 3. Ponds 2 and 3 have nitrogen limitations inducing lipids production. E) The pH of the system ranges between 5 and 7.

simulated the case where yeast can metabolize both glucose and xylose.

Figure 4-12 presents a schematic of the cultivation system. Cellulosic glucose and xylose are metabolized by yeast and converted to CO₂ which is then fixed by algae. A total of 62.9, 31.5, 2.8, and 0.03 g/(m² × day) of glucose, xylose, ammonia, and sodium hydroxide, respectively, are fed into the system. Figure 4-12 shows how these feeds are distributed among the three ponds; the last pond is nitrogen-limited, and

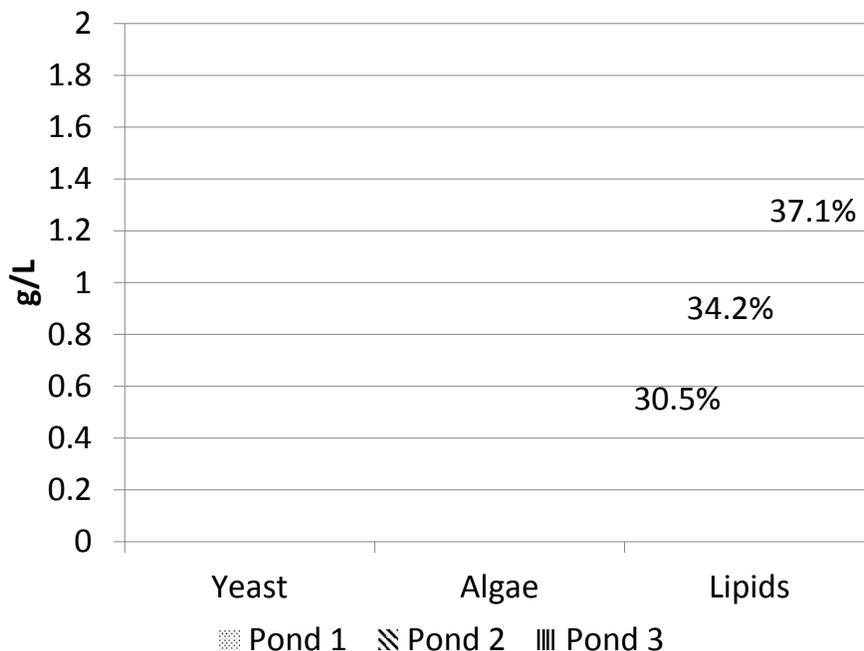


Figure 4-11: Yeast, algae and lipids concentrations in a cultivation system using three raceway ponds with cellulosic glucose feed. The numbers on top of the Lipids bars represent the weight fraction of lipids in total biomass. The last two ponds are nitrogen-limited; therefore, the model predicts significant lipids accumulation (up to $\approx 37.1\%$ weight). Yeast grows slower in the last two ponds due to nitrogen limitations and lower glucose feed rates.

lipids production is induced.

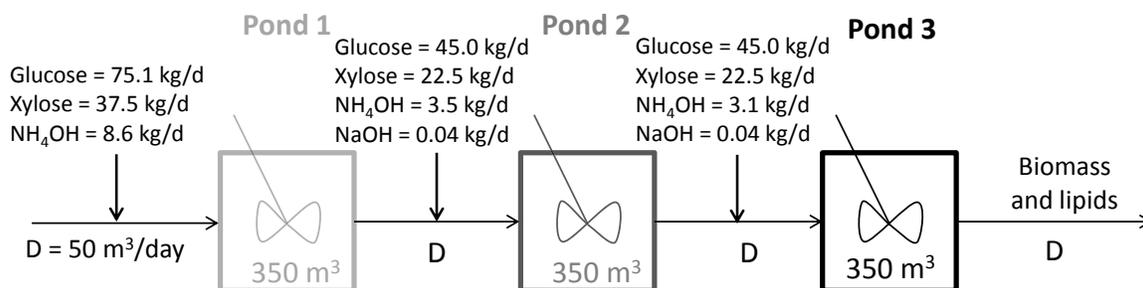


Figure 4-12: Schematic of the algal biomass cultivation system using three raceway ponds and cellulosic glucose and xylose feeds. Each pond can be modeled as a CSTR with a volume of 350 m^3 . There is a constant feed and outlet of $50 \text{ m}^3/\text{day}$ for each pond. Sodium hydroxide is fed at a constant rate all 24 hours a day, whereas glucose, xylose and ammonia are fed only during daytime (12 hours).

Figure 4-13 shows that this cultivation scheme can attain yeast, algae, and lipids

productivities of 30.2, 27.4 and 22.9 g/(m² × day), respectively. Yeast and algae accumulate lipids up to approximately 40% and 39% dry weight, respectively. Figure 4-14 shows algae, yeast and lipids concentrations in each pond. Due to nitrogen limitations in the last two ponds, lipids are accumulated. Table 4.7 presents the carbon balance for this case; approximately 80.6% of the carbon in glucose and xylose ends in biomass.

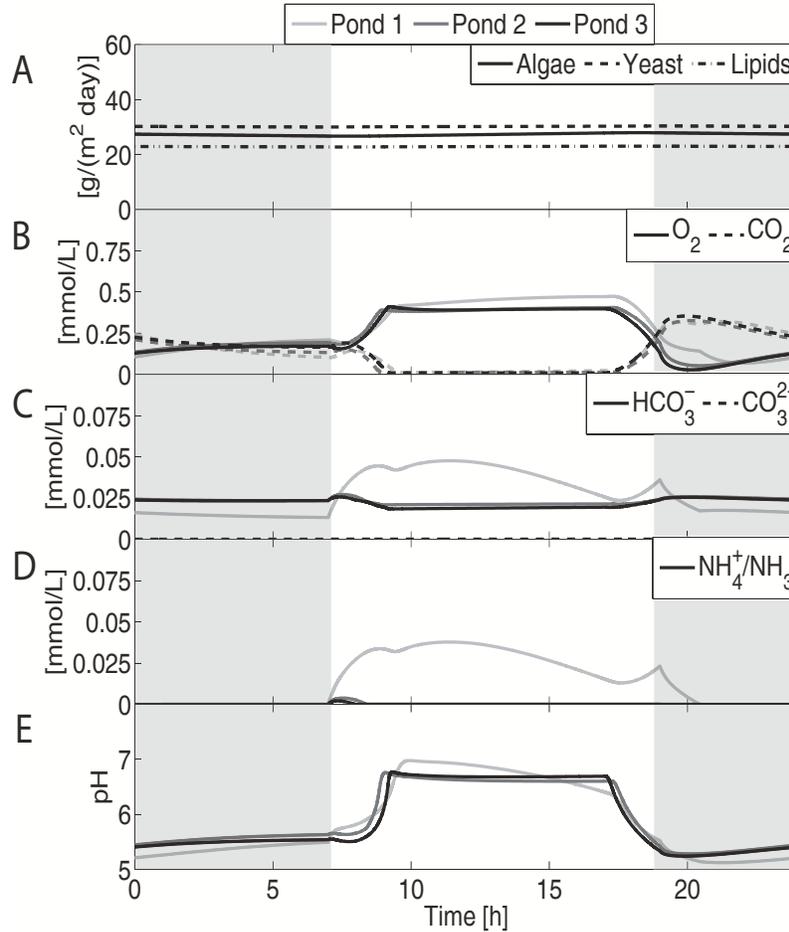


Figure 4-13: Concentration profiles of an algae/yeast cultivation system using three raceway ponds with cellulosic glucose and xylose feeds. Shaded areas represent dark periods. A) Predicted yeast, algae, and lipids productivities are approximately 30.2, 27.4 and 22.9 g/(m² × day), respectively. B) Due to algae's photosynthetic activity, O₂ concentration increases during the day and decreases during the night, whereas CO₂ concentration decreases during the day and increases during the night. C) HCO₃⁻ and CO₃²⁻ concentrations remain low. D) The concentration of nitrogen drops as we move from Pond 1 to Pond 3. Ponds 2 and 3 have severe nitrogen limitations inducing lipids production. E) The pH of the system ranges between 5 and 7.

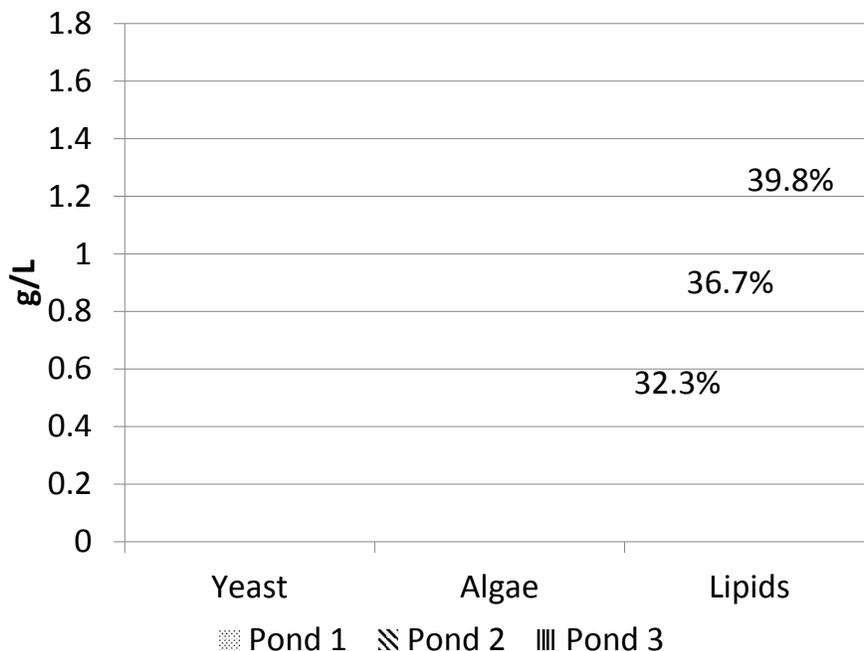


Figure 4-14: Yeast, algae and lipids concentrations in an algae/yeast cultivation system using three raceway ponds with cellulosic glucose and xylose feeds. The numbers on top of the Lipids bars represent the weight fraction of lipids in total biomass. The last two ponds are nitrogen-limited; therefore, the model predicts significant lipids accumulation (up to $\approx 39.8\%$ weight). Yeast grows slower on the last two ponds due to nitrogen limitations and lower glucose and xylose feed rates.

Table 4.7: Carbon balance of coculture with glucose/xylose feed.

Carbon Inputs		Carbon Outputs	
Glucose	66.7%	Yeast biomass	45.0%
Xylose	33.3%	Algal biomass	35.7%
		Net loss to atmosphere	19.2%
		Inorganic carbon lost in flow	0.1%
		Xylose lost in flow	0.1%
		Glucose lost in flow	0.01%

4.2.6 Algae/yeast coculture with cellulosic glucose and xylose feeds with acetate production

In the previous alternative, the feeds of sugars were kept low to avoid acetate production. However, if more sugars are fed into the system, a higher productivity of yeast biomass can be attained and the acetate produced can be metabolized by algae. Although some efficiency is lost in the carbon balance, a lower biodiesel price can be

attained as the impact of capital costs is reduced. In this alternative, sugars were fed all day long with extra sugars being fed during daytime. Figure 4-15 shows the sugars, caustic soda, and ammonia feed distribution.

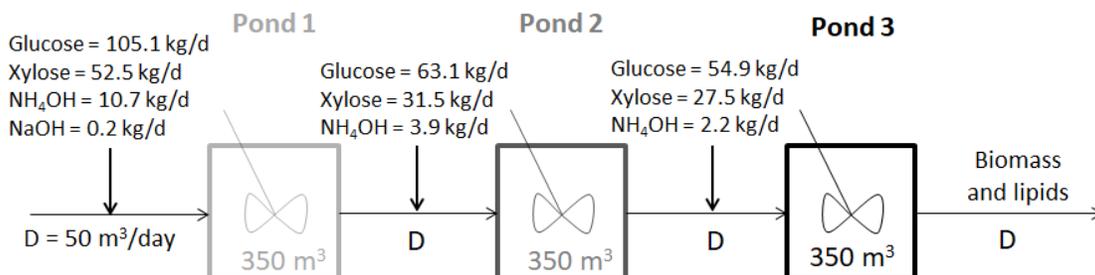


Figure 4-15: Schematic of the algal biomass cultivation system using three raceway ponds and cellulosic glucose and xylose feeds with acetate production. Each pond can be modeled as a CSTR with a volume of 350 m³. There is a constant feed and outlet of 50 m³/day for each pond. Sodium hydroxide is fed at a constant rate all 24 hours a day, whereas glucose, xylose and ammonia are fed at a higher rate during daytime (12 hours).

Figure 4-16 shows the results of this cultivation method. It can be observed that higher productivities are attained compared to the alternative suppressing acetate production. In addition, the pH is still between 5 and 7, the concentrations of oxygen and carbon dioxide show that photosynthesis is occurring, and the last two ponds are nitrogen-limited. In this particular cultivation alternative it was noted that the last pond was not as productive in lipids or biomass accumulation as the first two ponds. Therefore, a system considering just the first two ponds was considered. The carbon balance of these two alternatives is presented in Table 3. In particular, productivities of algae biomass, yeast biomass and lipids go from 26.2, 39.5, and 30 g/(m² day), respectively with a three pond system, to 27.2, 51.2, and 31.9 g/(m² day), respectively with a two pond system.

The carbon balance in Table 4.8 shows that about 70-74% of the carbon in the sugars is fixed as biomass, which is lower than the 80% observed in the previous cultivation alternative. This is a result of both algae and yeast growing heterotrophically at night in addition to algae's heterotrophic growth on acetate. Therefore, more CO₂ is generated.

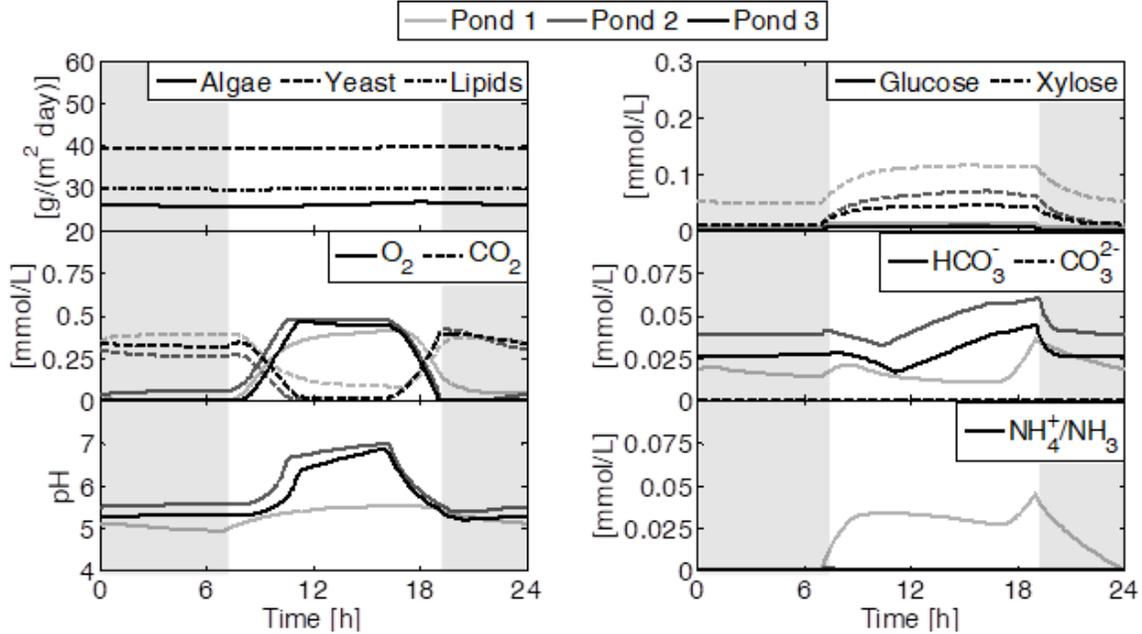


Figure 4-16: Concentration profiles of an algae/yeast cultivation system using three raceway ponds with cellulosic glucose and xylose feeds with acetate production. Shaded areas represent dark periods. A) Predicted yeast, algae, and lipids productivities are approximately 39.5, 26.2 and 30 $\text{g}/(\text{m}^2 \times \text{day})$ for a three pond system, respectively. B) Due to algae's photosynthetic activity, O_2 concentration increases during the day and decreases during the night, whereas CO_2 concentration decreases during the day and increases during the night. C) HCO_3^- and CO_3^{2-} concentrations remain low. D) The concentration of nitrogen drops as we move from Pond 1 to Pond 3. Ponds 2 and 3 have severe nitrogen limitations inducing lipids production. E) The pH of the system ranges between 5 and 7.

Table 4.8: Carbon balance of a coculture with glucose/xylose feed and acetate production.

Carbon Inputs		Carbon Outputs	
		3 ponds	2 ponds
Glucose	66.7%	Yeast biomass	43.5%
Xylose	33.3%	Algal biomass	26.8%
		Net loss to atmosphere	29.4%
		Total carbon lost in flow	0.3%
			0.3%

Biomass cultivation systems based on lignocellulosic sugars, as the ones illustrated in Sections 4.2.4, 4.2.5, and 4.2.6, can be implemented at locations that generate large amounts of cellulosic waste, such as farms or forests. The area suitable for this scheme is much larger than the one offered by the scheme requiring flue gas in Section 4.2.2.

4.2.7 Economic Analysis

We now proceed to do a simple economic analysis based on Table 9 of Williams and Laurens [143]. First we describe the assumptions for this economic analysis.

Raceway Pond Costs

The capital costs for a raceway pond were estimated from Chisti [24]. A plastic-lined pond including earth works, carbon dioxide supply piping, inlets, outlets, baffles, paddlewheel, and motor had a cost of \$69,500/hectare in 1987. Using CPI indices, this translates to roughly \$146,000/hectare in 2015 dollars.

Each pond has a paddlewheel running for 24 hours. The power consumed by the paddlewheel can be calculated using the following equation [24]:

$$P = \frac{1.59A\rho g u^3 f_M^2}{e d_h^{0.33}} \quad (4.8)$$

where P is the power in Watts, $A = 850 \text{ m}^2$, $\rho \approx 1000 \text{ kg/m}^3$ is the density of the culture broth, $g = 9.81 \text{ m/s}^2$, $u = 0.23 \text{ m/s}$ is the flow velocity, $f_M = 0.015 \text{ s/m}^{1/3}$ is the Manning channel roughness factor for concrete, e is the paddlewheel efficiency estimated at about 0.17 [24], and $d_h = 0.96 \text{ m}$ is the hydraulic diameter of the channel. This gives an estimate of $216.4 \text{ W/pond} = 5.19 \text{ kWh/pond} \times \text{day}$.

Flue Gas Compression

Let us assume that flue gas is available for free. An estimated pressure drop of 1 bar/km is predicted [132]. Let us suppose that compression is done adiabatically, with compression ratio equal to 4, and efficiency of 80% [132]. From McCabe *et al.* [91], the following equation can be used to calculate the power requirements of compression:

$$Power = \frac{0.371 T_a \gamma q_0}{(\gamma - 1) \eta} \left[\left(\frac{P_{out}}{P_{in}} \right)^{1-1/\gamma} - 1 \right] \quad (4.9)$$

with power in kW, q_0 the flowrate in m^3/s at $0 \text{ }^\circ\text{C}$ and 1 atm, $T_a = 298\text{K}$ the inlet temperature and η the efficiency. To obtain γ , the composition of flue gas was

assumed to be 81.4% N₂, 13.6% CO₂ and 5% O₂ [17]. Their heat capacities at 20 °C are 29.1, 36.9, and 29.3 J/mol×K, respectively, resulting in an average $\gamma = 1.386$. A flowrate of 1 m³/h at 298 K is equivalent to 2.545×10^{-4} m³/s at 273 K. Multistage compression is used when the desired compression ratio is higher than 4.

We estimated the capital costs of a centrifugal compressor with maximum capacity of 30,000 kW using the correlations in Table 7.2 of Towler and Sinnott [129]. We calculated the compressor capital costs considering a cultivation area of 900 hectares. We considered that the compressor was made of stainless steel due to the acids present in the flue gas, and used the factors in Tables 7.4 and 7.5.

Raw Materials

The price of cellulosic glucose is \$0.126/kg [5], and that of a mix of glucose and xylose is \$0.07/kg [113]. From Table 4 in Davis, the price of ammonia is \$0.407/kg and of diammonium phosphate (DAP) is \$0.442/kg [29]. From ICIS Chemicals, the cost of NaOH was \$0.83/kg [62]. The cost of methanol and natural gas were obtained from Williams and Laurens [143].

Other Considerations

The density of biodiesel is 0.88 kg/L, and it is assumed that 70% of biolipid production is equal to the production of biodiesel [143] due to losses in the esterification reaction and downstream processing inefficiencies. It is assumed that 0.3 cm/day corresponding to 30 L/day/hectare of water are lost to evaporation, that all water in the outlet stream can be recycled, and that the cost of water is \$0.05/1000 gallon [29]. Capital and operating costs associated to engineering, harvest, extraction, anaerobic digestion, and labor were obtained from Williams and Laurens considering a price of electricity of 10.2 cents/kWh [135]. Only 12% of the site preparation costs reported in Table 9 of Williams and Laurens were maintained in the analysis presented in this chapter, as 88% of the site area is devoted to the ponds and the site preparation costs for the pond are already considered in the pond capital cost estimate [143]. The capital costs are divided by 10 years of operation [143]. It is assumed that carbohydrates

and proteins are processed in an anaerobic digester (AD), and their calorific values (17.3 and 23.9 kJ/g, respectively) are transformed to electricity with an efficiency of 26.25% [143]. A ratio of 3:2 protein to carbohydrates is assumed [143].

In the economic analysis of Williams and Laurens, protein is sold as animal feed at \$0.9/kg. At this price, the income from protein exceeds the one obtained from biodiesel. If microalgal ponds are used for large-scale production of biodiesel, the market for animal feed will be flooded and the price of protein as animal feed will fall. Therefore, in this analysis we assumed that all protein was digested anaerobically. This allows to recover nitrogen and phosphorus such that 60% of the nutrient inputs to the process are recycled [24].

An open pond system is prone to contamination and invasion by other species. In fact, in some cases contamination can cause the desired algae strain concentration to fall by approximately 90% in a matter of days [106]. Some strategies such as adding toxic chemicals or increasing salt concentration to kill potential invaders, constant feeding of the desired algal and yeast strains, and growing algae and yeast in coculture reduce this risk. To consider contamination and invasion risks, the final biodiesel yield was divided by two.

First, the economic analysis of the sparging alternative is presented. From Table 4.9 we notice that the monoculture is very competitive when it is very close to a source of flue gas, but its price increases rapidly with distance. High sparging rates are desirable because they attain higher concentrations of biomass, making the capital costs less burdensome. However, as distance from a flue gas source increases, higher sparging rates incur higher operating costs.

Next we present the results for a coculture growing on pure glucose, one growing on a mix of glucose and xylose, and a yeast monoculture growing on a mix of glucose and xylose (Table 4.10). The benefits of using a mix of sugars instead of pure glucose is evident as the price decreases approximately $\approx 30\%$ with respect to pure glucose. The yeast monoculture performs slightly better than the coculture, although the difference is probably within the accuracy of our cost estimates. There are several benefits of using a coculture with respect to a monoculture:

Table 4.9: Economic analysis for biodiesel production using CO₂ sparging.

Sparging rate (m³/h)	10	40	100	Units
Areal Production	11.7	34.4	48.8	gDW/(m ² × d)
	42.7	125.6	178.1	tonne/(ha × yr)
Lipid Content	47.4	47.1	46.9	%DW
Lipid Production	20.2	59.1	83.6	tonne/(ha × yr)
Carbohydrate Prod.	9.0	26.6	37.8	tonne/(ha × yr)
Protein Prod.	13.5	39.9	56.7	tonne/(ha × yr)
Electricity AD	34.8	103	146.5	MWh/(ha × yr)
Capital Costs				
Raceway Pond	146	146	146	\$1000/ha
Non-pond Site Prep.	5	5	5	\$1000/ha
Engineering	22	22	22	\$1000/ha
Harvest	3.6	3.6	3.6	\$1000/ha
Extraction	4.6	4.6	4.6	\$1000/ha
Anaerobic Digester	5.1	14.9	21.2	\$1000/ha
TOTAL CAPITAL COSTS	186.2	196.1	202.4	\$1000/ha
Power Requirements				
Growth	21.6	21.6	21.6	MWh/(ha × yr)
Harvest	38	38	38	MWh/(ha × yr)
Dewatering	1.5	1.5	1.5	MWh/(ha × yr)
Electricity Required	26.3	-41.8	-85.4	MWh/(ha × yr)
Running Costs				
Labor	4430	4430	4430	\$/ (ha × yr)
Electricity	2640	-4191	-8557	\$/ (ha × yr)
Transesterification				
Power (Natural Gas)	93.2	272	385	\$/ (ha × yr)
Methanol	927	2710	3830	\$/ (ha × yr)
NaOH	462	462	462	\$/ (ha × yr)
Water	145	145	145	\$/ (ha × yr)
Ammonia	69.6	207	295	\$/ (ha × yr)
DAP	586.2	1740	2480	\$/ (ha × yr)
Anaerobic Digester	1744	5160	7350	\$/ (ha × yr)
Capital Costs @ 10%/yr	18.6	19.6	20.2	\$1000/(ha × yr)
TOTAL COSTS	29.7	30.5	31.1	\$1000/(ha × yr)
Biodiesel Prod. Ideal	16.1	47.0	66.5	kL/(ha × yr)
Biodiesel Prod. Real	8.0	23.5	33.2	kL/(ha × yr)
Biodiesel Cost (0 km)	3.70	1.30	0.93	\$/L
Biodiesel Cost (1 km)	4.08	1.66	1.54	\$/L
Biodiesel Cost (3 km)	4.36	1.98	2.09	\$/L
Biodiesel Cost (15 km)	5.02	2.66	3.25	\$/L
Biodiesel Cost (63 km)	5.68	3.34	4.41	\$/L

1. In a coculture, resources are better utilized making invasion more difficult. A coculture is more resilient than a monoculture [70].

2. If CO₂ generation is a concern, the coculture releases less CO₂ to the atmosphere than a yeast monoculture as algae metabolizes CO₂. In the monoculture 37 % of the carbon in the sugars is released as CO₂ compared to 19% in the coculture.
3. The coculture utilizes less raw materials. In mature commodity industries, raw materials costs represent approximately 70% of the total cost; therefore, reducing raw materials consumption is important.

Therefore, an algae/yeast coculture could be preferred over a yeast monoculture.

Finally, Table 4.11 compares the systems using sugar mix as a feed with and with no acetate production. It can be observed that when acetate is produced, lower biodiesel prices can be attained in the order of \$1.31/litre for the three pond alternative and \$1.26/litre for the two pond alternative. Again, half of the yield is assumed to be lost due to invading species. It is important to notice that the downstream costs have not been modified to account for a more diluted output from the two pond alternative compared to the three pond one.

The anaerobic digester produces CO₂. If algae ponds are located immediately next to the digester, minimal costs are incurred for compressing the flue gas. Then, the price per liter of biodiesel for a sparging rate of 100 m³/h is of \$0.93/L. A mixed setup can be conceived where most of the biodiesel is produced from a glucose/xylose mix and a few ponds produce biodiesel using flue gas from the anaerobic digester.

4.3 Conclusions

DFBA can be used to model accurately complex and novel biological scenarios, for example, a microbial consortia in an algal pond. Based on this modeling framework, the potential of producing biodiesel in raceway ponds from algae and oleaginous yeast was evaluated. Flue gas can be used to produce biodiesel at competitive prices only if the ponds are located very close to the flue gas source. Meanwhile, algae/yeast cocultures provide a method of producing biodiesel using cellulosic sugar. Our model predicts a cost of production of biodiesel of \$2.01/L if pure glucose is used and of

Table 4.10: Economic analysis for biodiesel production with pure glucose and a glucose/xylose mix.

	Cocul.	Cocul.	Yeast	Units
Glucose or Mix	Gluc.	Mix	Mix	
Yeast Production	34.5	30.2	86.9	gDW/(m ² × d)
Algae Production	26.2	27.3	0	gDW/(m ² × d)
Lipid Production	82.5	83.6	128.7	tonne/(ha × yr)
Carbohydrate Prod.	55.6	50.5	75.5	tonne/(ha × yr)
Protein Prod.	83.4	75.8	113.2	tonne/(ha × yr)
Electricity AD	216	196	292	MWh/(ha × yr)
Capital Costs				
Raceway Pond	146	146	146	\$1000/ha
Non-pond Site Prep.	5	5	5	\$1000/ha
Engineering	22	22	22	\$1000/ha
Harvest	3.6	3.6	3.6	\$1000/ha
Extraction	4.6	4.6	4.6	\$1000/ha
Anaerobic Digester	31.2	28.4	42.4	\$1000/ha
TOTAL CAPITAL COSTS	212.4	209.6	223.6	\$1000/ha
Power Requirements				
Growth	21.6	21.6	21.6	MWh/(ha × yr)
Harvest	38	38	38	MWh/(ha × yr)
Dewatering	1.5	1.5	1.5	MWh/(ha × yr)
Electricity Required	-154.4	-134.6	-231.3	MWh/(ha × yr)
Running Costs				
Labor	4430	4430	4430	\$(ha × yr)
Electricity	-15400	-13500	-23200	\$(ha × yr)
Transesterification				
Power (Natural Gas)	380	385	593	\$(ha × yr)
Methanol	3780	3830	5900	\$(ha × yr)
NaOH	90.9	90.9	0	\$(ha × yr)
Water	145	145	145	\$(ha × yr)
Ammonia	1300	1150	2780	\$(ha × yr)
DAP	2360	2190	1690	\$(ha × yr)
Sugars	43400	24100	50400	\$(ha × yr)
Anaerobic Digester	4320	3930	5870	\$(ha × yr)
Capital Costs @ 10%/yr	21.2	21.0	22.4	\$1000 / (ha × yr)
TOTAL COSTS	65.8	47.6	70.8	\$1000 / (ha × yr)
Biodiesel Prod. Ideal	65.6	66.5	102.4	kL/(ha × yr)
Biodiesel Prod. Real	32.8	33.2	51.2	kL/(ha × yr)
Biodiesel Cost	2.01	1.44	1.39	\$/L

\$1.44/L if a mix of glucose and xylose is used instead when no acetate is produced. If we are willing to pay higher operating costs and have a higher carbon loss to the atmosphere by allowing acetate to be produced, the price can be reduced even further

Table 4.11: Economic analysis comparison between coculture systems for biodiesel production using a glucose/xylose mix. NAP: no acetate production, AP2: acetate production for a 2 pond system, AP3: acetate production for a 3 pond system.

	NAP	AP3	AP2	Units
Yeast Production	110.5	186.9	144.2	tonne/(ha × yr)
Algae Production	99.2	99.4	95.6	tonne/(ha × yr)
Lipid Production	83.6	116.6	109.5	tonne/(ha × yr)
Electricity AD	196	263	202	MWh/(ha × yr)
Capital Costs				
Raceway Pond	181.1	181.1	181.1	\$1000/ha
Anaerobic Digester	28.4	38.2	29.3	\$1000/ha
Operating Costs				
Electricity	-13,500	-20,200	-14,100	\$/ (ha × yr)
Power (Natural Gas)	385	537	505	\$/ (ha × yr)
Labor	4,430	4,430	4,430	\$/ (ha × yr)
Raw Materials	31,500	46,900	41,200	\$/ (ha × yr)
Anaerobic Digester	3,930	5,280	4,050	\$/ (ha × yr)
Operating Costs				
Capital Costs @ 10%/yr	21.0	21.9	21.0	\$1000 / (ha × yr)
TOTAL COSTS	47.6	58.8	57.1	\$1000 / (ha × yr)
Biodiesel Prod. Real	33.2	46.3	43.6	kL/(ha × yr)
Biodiesel Cost	1.44	1.27	1.31	\$/L

to \$ 1.26/L of biodiesel when a mix of glucose/xylose is fed into the system.

The results of this work suggest that algae/yeast cocultures for biodiesel production should be considered seriously. This alternative employs cellulosic sugars which are currently very cheap. In this analysis we considered that lipid-extracted biomass was utilized to produce electricity. Another option would be to treat it and make it digestible by the consortia, potentially reducing the operating costs of the consortia alternative. In addition, the results in this work are not systematically optimized. The optimization of this system requires the computation of generalized derivatives for non-smooth objective functions. The work in Khan *et al.* [72], Höffner *et al.* [60], and [47] enables the numerical optimization of these systems as described in Chapters 6,7, and 8 of this thesis. However, despite the lack of systematic optimization in this example, the results of the algae/yeast coculture growing on cellulosic sugars presented in this chapter are promising. We suggest experimental groups implement the proposed microbial consortia strategy to increase culture resilience and expand the range of substrates that can be converted into biofuels.

Chapter 5

Multispecies Raceway Pond Modeling

The work in this chapter is based on the work published in [45] and in the conference proceedings [8] and builds upon the model presented in Chapter 4. Microbial consortia provide numerous benefits such as culture resilience, better resource utilization and symbiosis. In this chapter, more species are considered as part of a consortia involving oleaginous microalgae, yeasts and bacteria, as a way of using both sunlight and lignocellulosic waste to grow biomass. The resulting dynamic flux balance analysis (DFBA) [137, 87] models were implemented in DFBAlab [44].

5.1 Materials and methods

The model described in this chapter builds upon the one presented in Chapter 4, which is based on the high-rate algal-bacterial pond (HRAP) model [18, 144] and DFBA.

5.1.1 Metabolic Network Reconstructions

This work considers four different microorganisms: bacterium *E. coli*, microalga *C. reinhardtii*, and yeasts *S. cerevisiae* and *R. glutinis*. The following GENREs were

used:

1. iJR904 [107]: This model for *E. coli* considers 2 compartments, 1075 reactions, and 761 metabolites.
2. iND750 [31]: This model for *S. cerevisiae* is divided into eight compartments and considers 1266 reactions and 1061 metabolites. This model correctly predicts ethanol production under anaerobic conditions. However, *S. cerevisiae* is not oleaginous. In this chapter we modified the iND750 model such that it cannot produce ethanol, it can consume xylose, and it can accumulate triglycerides, approximating the behavior of oleaginous yeast *R. glutinis* [146].
3. iRC1080 [19]: This model for *C. reinhardtii* is divided into ten compartments and considers 2191 reactions and 1706 metabolites out of which 125 were classified as lipids. For each one of these lipids, a storage reaction was added to the model. The model predictions have been validated experimentally for nitrogen-limited and light-limited growth.

For all lipids storages, a production energy requirement of two times the energy required for normal biomass synthesis on a per weight basis was imposed.

Figure 5-1 illustrates the possibilities for symbiosis and competition amongst the different microorganisms in the consortium.

5.1.2 The Raceway Pond Model

For more details refer to Section 4.1.3. This chapter considers series of two or three raceway ponds open to the atmosphere of the same size and operating parameters as in Chapter 4. In addition, the same average surface light intensity of $18.81 \text{ MJ}/(\text{m}^2 \times \text{day})$ is considered and is modeled according to Equations (4.1), (4.2) and (4.3).

In the same way as in Chapter 4, Michaelis-Menten expressions were used as upper bounds on the exchange fluxes:

$$v_S^U = \left(\frac{v_{max}S}{K_m + S} \right) \left(\frac{1}{1 + I/K_I} \right) \beta \quad (5.1)$$

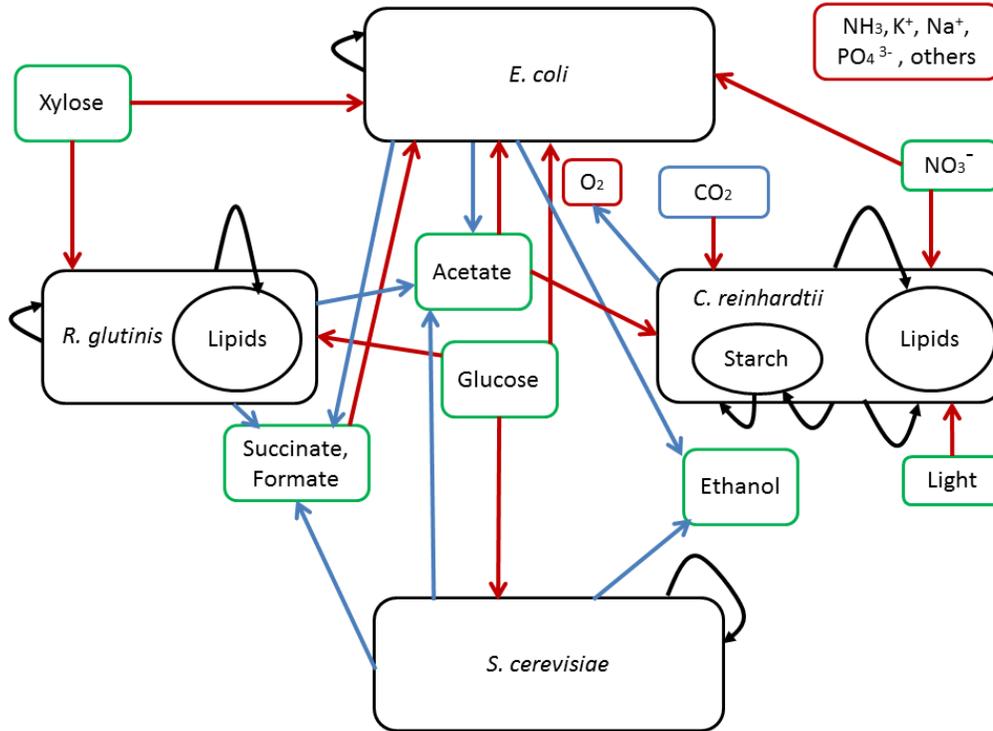


Figure 5-1: Schematic of the interactions amongst *E. coli*, *C. reinhardtii*, *S. cerevisiae*, and *R. glutinis*. The black rectangles refer to microorganisms and their internal storages. Black arrows refer to biomass, lipids, and starch fluxes. Blue arrows refer to metabolic products, whereas red arrows refer to uptakes from extracellular substrates and nutrients. Green rectangles refer to substrates and products that are just specific to a subset of microorganisms. The rectangles referring to O_2 and nutrients are marked as red as all microorganisms consume these species, whereas CO_2 is marked as blue as all microorganisms have it as a metabolic product. It is evident from this schematic that there are significant interactions among the different microorganisms in the consortium. A framework such as DFBA can explore and model these relationships and how they change with extracellular conditions.

where S is the substrate concentration, I is the inhibitor concentration, β is a pH factor and v_{max} , K_m and K_I are constants. The pH factor is important because algae prefer slightly basic pH whereas yeasts prefer acidic pH. The shape of the pH factors are given by Equation (4.5) and are illustrated in Figure 5-2 for the different species. Starch accumulation by algae follows Equations (4.6) and (4.7).

The values of the constants for Equation (5.1) are taken from the literature and presented in Tables 5.1 and 5.2. The uptake of acetate in algae was modeled according to Zhang using the expression for growth on ammonium chloride [147], but since this

expression is slightly different from Eq. (4.4), the values of its constants are not reported in Table 5.1. Table 5.2 presents the constants used in Equation (4.5) for

Table 5.1: Summary of uptake kinetic parameters for all microorganisms

	v_{\max} [$\frac{mmol}{gDW \times h}$]	K_m [mM]	K_I [mM]	Reference
<i>E. coli</i>				
Glucose	10.5	0.0027	EtOH: 435	Table I in [51]
Xylose	6	0.0165	EtOH: 435	Table I in [51]
O ₂	15	0.024	None	Table I in [51]
Acetate	3	0.0165	EtOH: 435	Assume 1/2 of xylose rate
Succinate	1	0.0165	EtOH: 435	Assume 1/6 of xylose rate
NH ₄ ⁺	10	0.1	None	Table I from [76], only K_m
<i>C. reinhardtii</i>				
CO ₂	1.25	0.03	None	Fig. 2 in [131]
O ₂	2.065	0.008	None	Table I in [144]
HCO ₃ ⁻	1.82	0.27	None	Fig. 2 in [131]
NH ₄ ⁺	0.65	3.84×10^{-4}	None	Fig. 1 and 2 in [57]
Acetate	N.A	N.A	None	Eq. (13) in [147]
<i>S. cerevisiae</i>				
Glucose	22.4	4.44	EtOH: 217	Table I in [51]
O ₂	2.5	0.003	None	Table I in [51]
NH ₄ ⁺	25.5	35.4×10^{-3}	None	Table I in [68]
<i>R. glutinis</i>				
Glucose	3.53	0.47	EtOH: 217 Xylose: 23.3	Tables 1 and 2 in [65]
Xylose	4.28	0.77	EtOH: 217 Glucose: 0.02	Tables 1 and 2 in [65]
O ₂	2.5	0.003	None	Table 1 in [51]
NH ₄ ⁺	25.5	35.4×10^{-3}	None	Table I in [68]

Notes: Ammonium uptake for yeasts was approximated with that of fungus *Lactarius rufus*. The weight fraction of chlorophyll (22.8 mg/gDW) in algae was obtained from the biomass equation in the iRC1080 model. The v_{\max} for *E. coli* uptake of NH₄⁺ is an assumed value.

these simulations.

The open ponds are in direct contact with the atmosphere, therefore a simple model based on film theory is used to estimate the mass transfer across the interface between air and water with parameters from Buhr and Miller [18] and Yang [144]. The equilibrium concentrations for both O₂ and CO₂ in water are calculated using Henry's law. The dissolved gas concentrations are limited by their saturation concentration at ambient conditions.

Table 5.2: Constants for pH dependent uptakes for different microorganisms

Species	Substrate	α [M]	K_1 [M]	K_H [M]	K_{OH} [M]	Note
<i>E. coli</i>	All substrates	1	1	10^{-3}	10^{-4}	(1)
<i>C. reinhardtii</i>	$\text{CO}_2/\text{HCO}_3^-$	1.08	1.07	5.9×10^{-6}	1.08×10^{-4}	(2)
<i>C. reinhardtii</i>	Acetate	1	1.1835	6.67×10^{-6}	3.54×10^{-7}	(3)
Yeasts	Glucose	1	1	10^{-3}	$10^{-4.8}$	(4)
<i>R. glutinis</i>	Xylose	1	1	$10^{-3.2}$	10^{-7}	(5)

Notes: (1) Fitted for pH range 3 - 11 from [32]. (2) Fitted from data in Fig. 3 in [79]. (3) Parameters from Eq. (13) in [147]. (4) Fitted from data in Table 2 in [65]. (5) Fitted from data in Table 2 in [65].

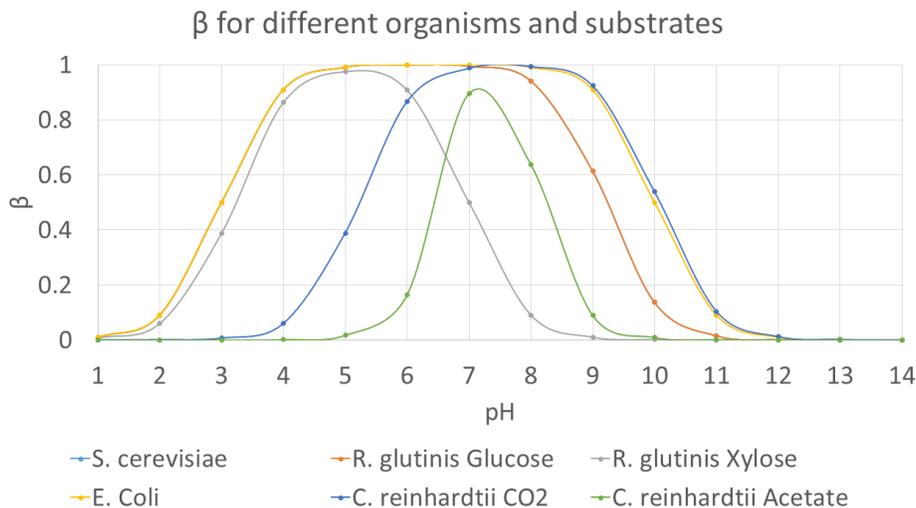


Figure 5-2: This plot shows the pH factor β for the different microorganisms in the consortia. For *C. reinhardtii* and *R. glutinis*, β has a different form for the different substrates. Notice that yeasts prefer acidic pH whereas the curves for *C. reinhardtii* are centered at around pH=8. *E. coli* has the widest range of pH suitable for growth.

The chemical equilibrium of the system is based on Buhr and Miller [18] with parameters obtained from Robinson and Stokes [109] at 20 °C. It is assumed that the ions present in the system are CO_3^{2-} , H^+ , OH^- , HCO_3^- , NH_4^+ , Na^+ , H_2PO_4^- , HPO_4^{2-} , PO_4^{3-} and the ions resulting from formic, acetic, and succinic acids. Ammonia and diammonium phosphate (DAP) are assumed to dissolve completely. Therefore, a system of equations is obtained from the solution equilibria of ammonia, carbon dioxide, formic, acetic, succinic, and phosphoric acids and water, the mass balances of ammonia, carbon, acetate, formate, succinate, and phosphate, and electroneutrality. From this system of equations, the concentrations of all ionic species are obtained.

5.1.3 Dynamic Flux Balance Analysis

As in Chapter 4, the raceway pond model and the GENREs are combined in a DFBA model which was implemented in DFBAlab [44]. The implementation of this process model in DFBAlab requires a hierarchy of objectives for each microorganism. The hierarchies of objectives used in this work are presented in Table 5.3. Notice that DFBAlab automatically enforces the first objective which is to minimize the penalty.

Table 5.3: Hierarchy of objectives used in DFBAlab.

	<i>E. coli</i>	<i>C. reinhardtii</i>
1	Maximize growth	Maximize autotrophic growth
2	Max/min ethanol production	Maximize lipids production
3	Minimize glucose consumption	Maximize starch production
4	Minimize xylose consumption	Maximize NH_4^+ consumption
5	Maximize CO_2 production	Maximize HCO_3^- consumption
6	Minimize O_2 consumption	Maximize NO_3^- consumption
7	Minimize NH_4^+ consumption	Maximize consumption and minimize production of O_2
8	Minimize NO_3^- consumption	Minimize formate production
9	Minimize PO_4^{3-} consumption	Minimize ethanol production
10	Minimize consumption and maximize production of acetate	Minimize acetate production
11	Maximize formate production	Minimize H_2 production
12	Minimize K^+ consumption	Maximize HPO_4^- consumption
13	Minimize Na^+ consumption	Minimize Na^+ consumption
14	Minimize consumption and maximize production of succinate	Minimize succinate production
	<i>S. cerevisiae</i>	<i>R. glutinis</i>
1	Maximize growth	Maximize growth
2	Maximize ethanol production	Maximize lipids production
3	Maximize CO_2 production	Maximize CO_2 production
4	Maximize glucose consumption	Maximize glucose consumption
5	Maximize O_2 consumption	Maximize xylose consumption
6	Minimize NH_4^+ consumption	Maximize O_2 consumption
7	Minimize acetate production	Maximize NH_4^+ consumption
8	Minimize formate production	Minimize acetate production
9	Maximize PO_4^{3-} consumption	Minimize formate production
10	Maximize K^+ consumption	Minimize PO_4^{3-} consumption
11	Minimize Na^+ consumption	Maximize K^+ consumption
12	Minimize succinate production	Maximize Na^+ consumption
13		Minimize succinate production

Growth rates and exchange fluxes rates are obtained from FBA. When the FBA

model becomes infeasible, the associated microorganism is unable to fulfill the requirements associated with maintenance. Therefore, it enters a death cycle. DFBAlab provides a penalty function whose value is associated with how far from feasibility the FBA model is; therefore, the higher the value of the penalty, the farther away the model is from feasibility and the faster its death rate will be. As a first approximation, we use this penalty as a death rate for *E. coli*, given that its associated FBA model becomes infeasible at some time intervals.

5.1.4 Economic Analysis

A simple economic analysis is performed based on Table 9 of Williams and Laurens [143] and is described in Section 4.2.7. Here we assume that only 40% of the nutrient inputs to the process are recovered as opposed to 60% in Chapter 4. Similar to Chapter 4, the final productivity is divided by two to account for the invasion of undesirable microorganisms.

5.1.5 Key differences from work in [45] and Chapter 4

1. Heterotrophic organisms are allowed to become oxygen limited. In this way, *C. reinhardtii* and *E. coli* can use some of the fermentation products such as acetate (acetate is not produced in the work in [45]);
2. The penalty function is used as a death rate;
3. The lipids accumulation in *R. glutinis* is now a variable determined by the FBA problem and not a fixed fraction of biomass growth;
4. Phosphate balances have been added into the system;
5. More microorganisms have been included in the simulation. In particular, some simulations contain four different species at the start of the simulation.

In Section 5.2 we explore slightly different communities and their respective productivities.

5.2 Results

First a simulation considering all four species was carried out. From this simulation, it was noticed that for most setups, only one heterotrophic species can occupy the heterotroph niche, which is consistent with [70]. In particular, *E. coli* was the most dominant heterotroph, followed by *R. glutinis* and *S. cerevisiae*, respectively. A summary of the feeds of the three pond system in Figure 5-3 are presented in Table 5.4.

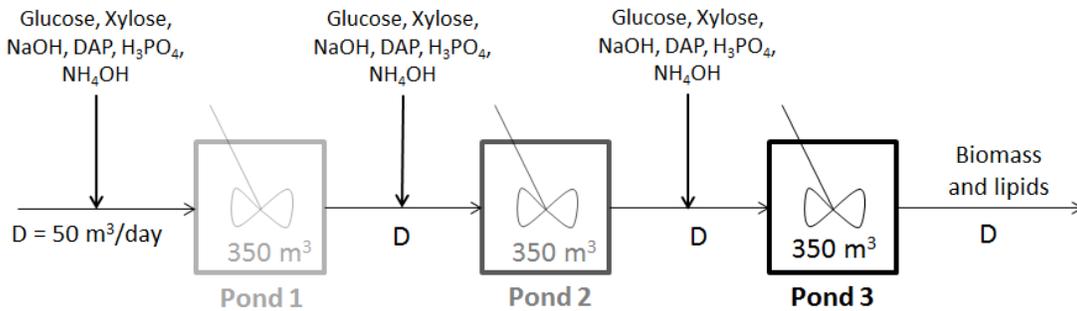


Figure 5-3: Three pond system for microbial cultivation: the first pond is meant to accumulate biomass whereas the last two ponds are meant for lipids accumulation.

5.2.1 Case 1: *C. reinhardtii* and *E. coli* coculture

In this case, it was assumed that *E. coli* maximizes ethanol production after biomass maximization (see Table 5.3). *E. coli* concentration is limited by the high ethanol concentration, which inhibits the uptake of sugars. From the carbon balance in Table 5.5 and Figure 5-10, it can be seen that most carbon in the process is converted into ethanol. In this case, no other heterotroph can survive in this environment. Under these conditions, a production cost of approximately \$3.54/L of biodiesel is predicted. Figure 5-4 shows the concentrations of products, substrates, and nutrients in the three-pond system.

Table 5.4: Feeds for the different pond distributions

	Pond 1		Pond 2		Pond 3	
	Day	Night	Day	Night	Day	Night
<i>C. reinhardtii/E. coli</i> coculture						
Glucose (kg/h)	4.07	2.81	4.07	2.81	4.07	2.81
Xylose (kg/h)	2.03	1.41	2.03	1.41	2.03	1.41
DAP (g/h)	289	289	0	0	0	0
NaOH (g/h)	5	5	13.3	13.3	13.3	13.3
NH ₄ OH (g/h)	540.2	248.2	0	0	0	0
H ₃ PO ₄ (g/h)	0	0	61.3	61.3	61.3	61.3
<i>C. reinhardtii/E. coli/R. glutinis</i> community						
Glucose (kg/h)	4.07	2.81	4.07	2.81	4.07	2.81
Xylose (kg/h)	2.03	1.41	2.03	1.41	2.03	1.41
DAP (g/h)	289	289	0	0	0	0
NaOH (g/h)	5	5	9.2	9.2	9.2	9.2
NH ₄ OH (g/h)	540.2	248.2	0	0	0	0
H ₃ PO ₄ (g/h)	0	0	44.9	44.9	44.9	44.9
<i>C. reinhardtii/R. glutinis</i> coculture						
Glucose (kg/h)	4.07	2.81	4.07	2.81	4.07	2.81
Xylose (kg/h)	2.03	1.41	2.03	1.41	2.03	1.41
DAP (g/h)	179	179	0	0	0	0
NaOH (g/h)	24.2	24.2	5.0	5.0	0	0
NH ₄ OH (g/h)	672	380	0	0	0	0
H ₃ PO ₄ (g/h)	0	0	30.1	30.1	20.4	20.4
<i>C. reinhardtii/S. cerevisiae</i> coculture						
Glucose (kg/h)	4.07	2.81	4.07	2.81	4.07	2.81
Xylose (kg/h)	2.03	1.41	2.03	1.41	2.03	1.41
DAP (g/h)	151	151	96.3	96.3	0	0
NaOH (g/h)	5.0	5.0	33.3	33.3	95.8	95.8
NH ₄ OH (g/h)	504	212	314	131	0	0
H ₃ PO ₄ (g/h)	0	0	0	0	40.8	40.8

5.2.2 Case 2: *C. reinhardtii*, *E. coli*, and *R. glutinis* cultivation.

If instead of maximizing ethanol production, *E. coli* minimizes ethanol production after biomass maximization (see Table 5.3), *R. glutinis* is able to survive in this environment. From the carbon balance in Table 5.5, most carbon is lost to the atmosphere, and from Figure 5-10 it can be seen that a higher lipids yield can be attained with respect to Case 1. Under these conditions, a production cost of approximately \$3.00/L of biodiesel is predicted. Figure 5-5 shows the concentrations of products,

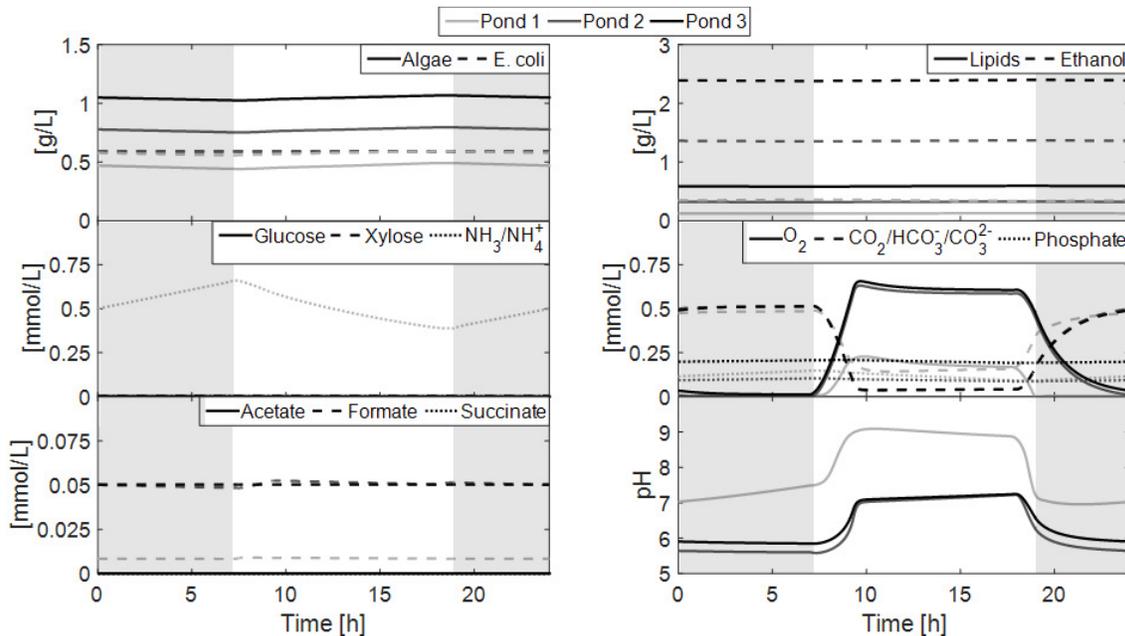


Figure 5-4: Concentrations of substrates, nutrients, and products in the three-pond system for Case 1. The dark areas of the plot represent nighttime. *C. reinhardtii* attains a higher biomass concentration than *E. coli*. The concentration of *E. coli* considers both live and dead cells. Due to the presence of *E. coli*, significant amounts of ethanol are produced. This setup uses substrates and nutrients efficiently; it can be seen that the last two ponds lack nitrogen, which is necessary for lipids production. The concentrations of oxygen and carbon dioxide change due to the photosynthetic activity of algae. Very little acetate, formate, and succinate are produced. The pH of the system stays between 5.5 and 9.

substrates, and nutrients in the three-pond system.

Notice that in this case, the model predicts that *R. glutinis* is able to accumulate up to 96 % lipids. This lipids fraction is unreal and underscores the need of a more detailed model that is able to take into account an upper limit for lipids accumulation.

5.2.3 Case 3: *C. reinhardtii* and *R. glutinis* coculture

If the cultivation system is kept free from *E. coli*, the most stable heterotroph is *R. glutinis*. This is the most desirable condition for biodiesel production. From the carbon balance in Table 5.5, 71.6% of carbon is converted into biomass. From Figure 5-10, a good lipids yield can be attained. Not surprisingly, from an economics standpoint this case results in the lowest production cost of approximately \$ 1.41/L

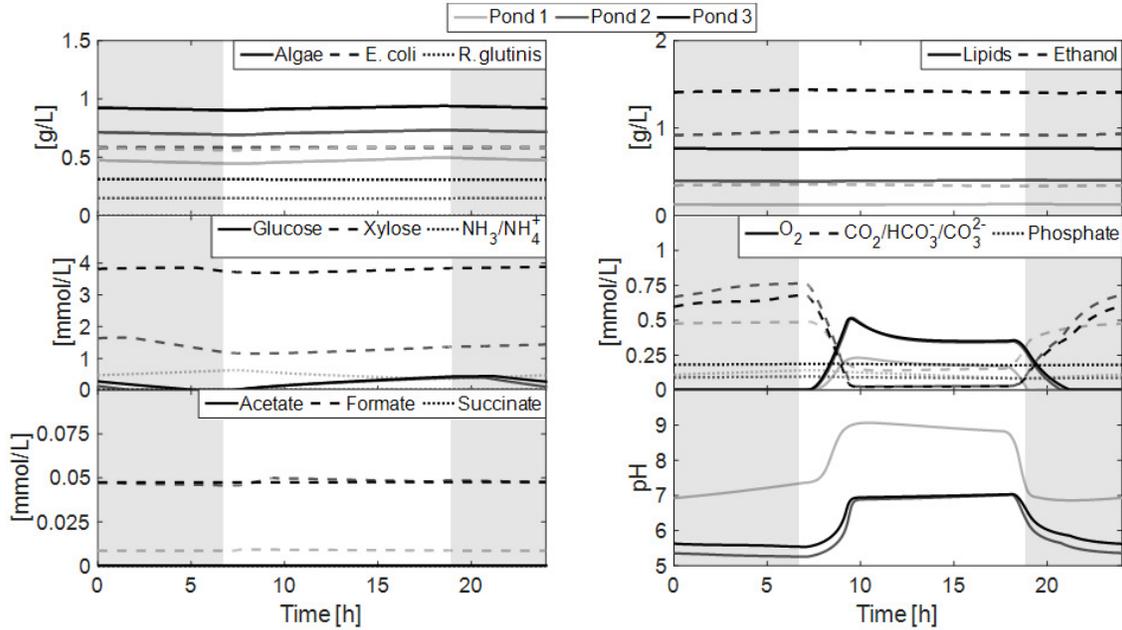


Figure 5-5: Concentrations of substrates, nutrients, and products in the three-pond system for Case 2. The dark areas of the plot represent nighttime. In this system *E. coli*, *C. reinhardtii*, and *R. glutinis* are all present. Significant amounts of ethanol are produced, although less than in Case 1. This setup does not use xylose efficiently; it can be seen that the last two ponds lack nitrogen, which is necessary for lipids production. The concentrations of oxygen and carbon dioxide change due to the photosynthetic activity of algae. Very little acetate, formate, and succinate are produced. The pH of the system stays between 5 and 9.

of biodiesel. This is the desired operating state for a cultivation system dedicated for biodiesel production. Figure 5-6 shows the concentrations of products, substrates, and nutrients in the three-pond system.

Case 1 and 2 are of interest because if the cultivation system were invaded by *E. coli*, we could expect the system to change towards Case 1 or 2. It can be seen that the addition of *E. coli* into the system can double the price of biodiesel. In particular, if a small concentration of 1 mg/L of *E. coli* appears in the first pond, Figure 5-7 shows the evolution of the pond concentrations if the inputs are not modified and Figure 5-8 shows how concentrations change if DAP and phosphoric acid inputs are modified to try to control the rise of pH. Modifying these inputs seem to have a very small effect in the system. It can be expected that the system will eventually migrate towards Cases 1 or 2. If a pH control system were used, the increase of *E. coli* and

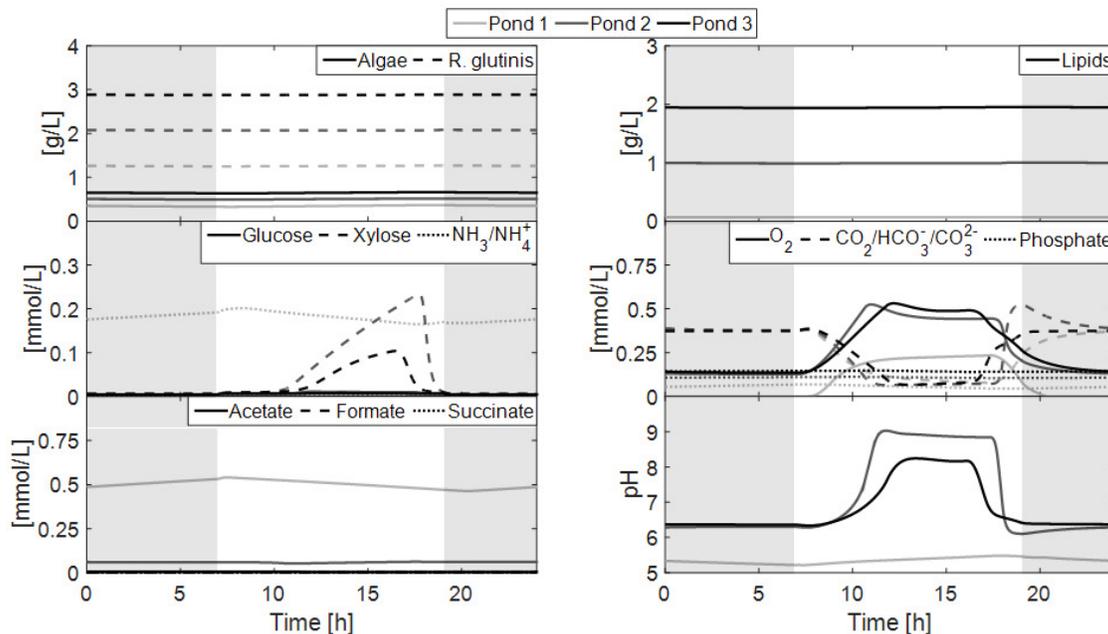


Figure 5-6: Concentrations of substrates, nutrients, and products in the three-pond system for Case 3. The dark areas of the plot represent nighttime. *R. glutinis* attains a higher biomass concentration than *C. reinhardtii*. Since *R. glutinis* is unable to produce ethanol, no ethanol is present in the system. This setup uses substrates and nutrients efficiently; it can be seen that the last two ponds lack nitrogen, which is necessary for lipids production. The concentrations of oxygen and carbon dioxide change due to the photosynthetic activity of algae. Some acetate is produced in the first pond, but it is consumed in the latter ponds. The pH of the system stays between 5 and 9.

ethanol concentrations could possibly be slowed down.

5.2.4 Case 4: *C. reinhardtii* and *S. cerevisiae* coculture

This case is not desired because of its low lipids productivity, nor expected because *S. cerevisiae* is less resilient than *E. coli* and *R. glutinis*. This results in a production cost of approximately \$ 5.74/L of biodiesel. Figure 5-9 shows the concentrations of products, substrates, and nutrients in the three-pond system.

A summary of the carbon balances is presented in Table 5.5. The carbon balances show that Case 3 is the most efficient in carbon utilization. A summary of the lipids fractions for all microorganisms in all ponds is presented in Table 5.6. It can be observed how nitrogen starvation promotes lipids accumulation up to 96% in the case

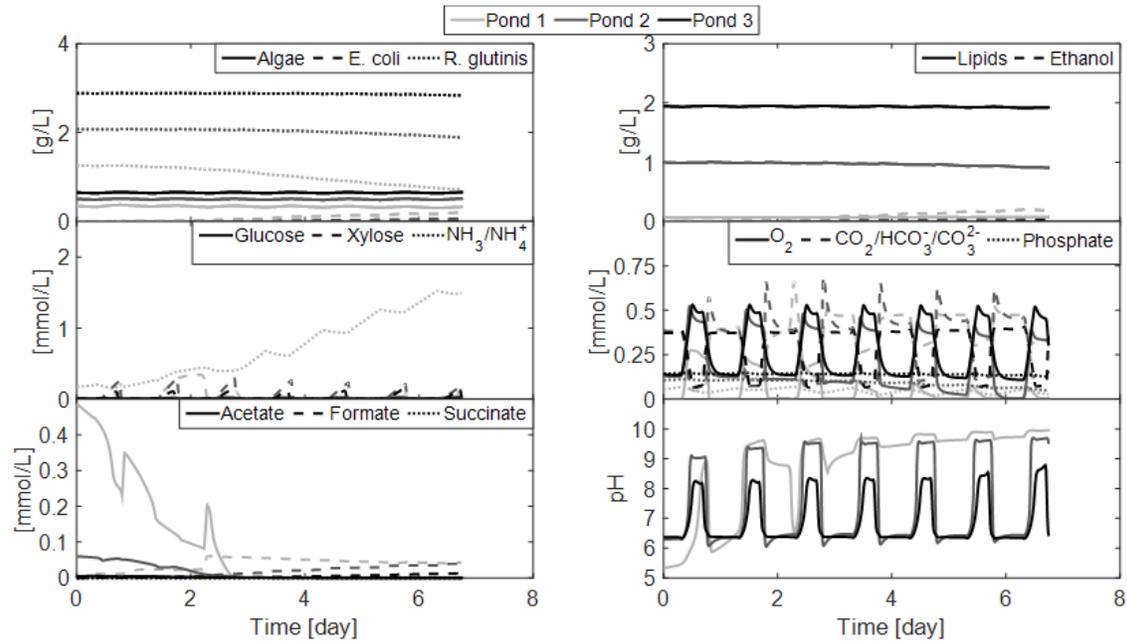


Figure 5-7: Evolution of concentrations after the appearance of 1 mg/L of *E. coli* in the first pond. A simulated time of slightly more than six days was used; the simulation was stopped when the system became a high-index DAE. The concentration of *E. coli* increases whereas the concentration of *R. glutinis* falls rapidly in the first pond. As the composition of the community changes, the concentration of nitrogen in the form of ammonia increases rapidly in the first pond causing an increase in pH. *E. coli* metabolizes the acetate in the first pond. Due to the presence of *E. coli*, some ethanol starts to appear in the system.

of *R. glutinis* and 49% in the case of *C. reinhardtii*. A summary of the costs for all four cases is presented in Table 5.7.

Table 5.5: Carbon balances for the different cases

	Case 1	Case 2	Case 3	Case 4
Biomass	27.1 %	31.5 %	71.6 %	36.8 %
<i>E. coli</i>	9.1 %	9.0 %	0	0
<i>C. reinhardtii</i>	18.0 %	14.7 %	11.2 %	14.7 %
<i>S. cerevisiae</i>	0	0	0	22.1 %
<i>R. glutinis</i>	0	7.8 %	60.4 %	0
Ethanol	41.9 %	24.9 %	0	0
Lost to atmosphere	30.9 %	35.3 %	28.2 %	28.9 %
Unconsumed substrates	0.01 %	8.2%	0.06 %	33.4 %
Other losses	0.13 %	0.15 %	0.11 %	1.0 %

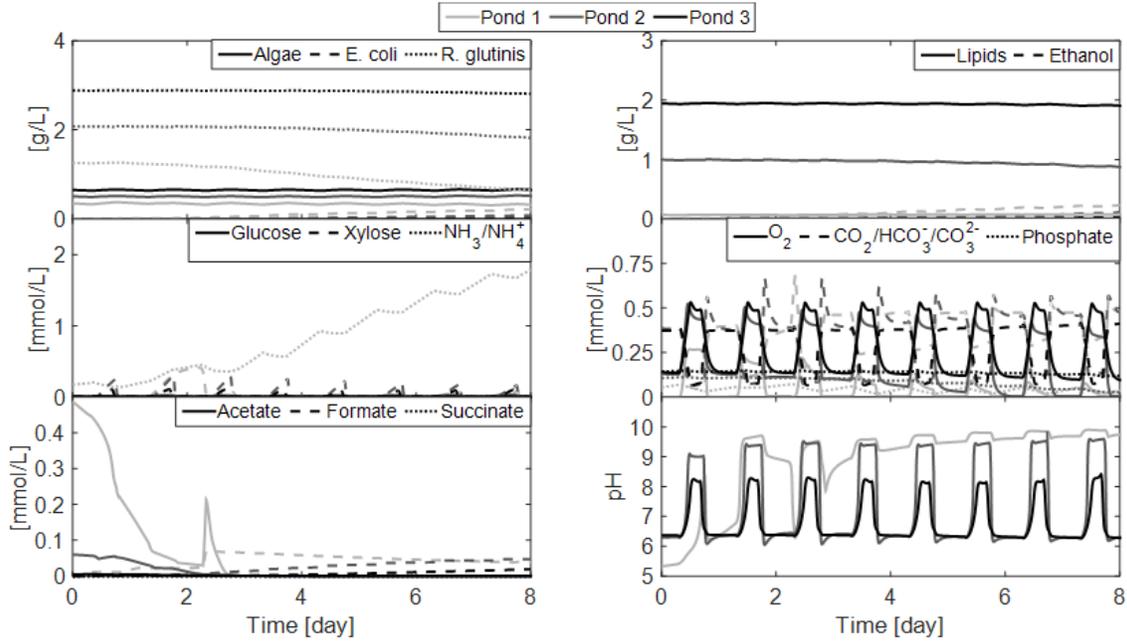


Figure 5-8: Evolution of concentrations after appearance of 1 mg/L of *E. coli* in the first pond with modified DAP and phosphoric acid inputs to control pH. Modifying these inputs seem to do little to change the rapid increase of pH, the increase of *E. coli* and ethanol in the system, and the rapid decrease of *R. glutinis*.

Table 5.6: Lipids fraction of biomass for each microorganism at each pond for all cases.

	Pond 1	Pond 2	Pond 3
Case 1			
<i>E. coli</i>	11.88 %	11.88 %	11.88%
<i>C. reinhardtii</i>	11.88 %	32.24 %	49.22 %
Total lipids fraction	11.88 %	23.45 %	35.78 %
Case 2			
<i>E. coli</i>	11.88 %	11.88 %	11.88%
<i>C. reinhardtii</i>	11.88 %	27.00 %	43.22 %
<i>R. glutinis</i>	NA %	91.73 %	95.96 %
Total lipids fraction	11.88 %	27.48 %	42.21 %
Case 3			
<i>C. reinhardtii</i>	11.80 %	34.25 %	48.43 %
<i>R. glutinis</i>	1.96 %	39.67 %	56.56 %
Total lipids fraction	4.09%	38.61%	55.07 %
Case 4			
<i>C. reinhardtii</i>	11.79 %	16.30 %	53.27 %
<i>S. cerevisiae</i>	1.96 %	1.96 %	1.96 %
Total lipids fraction	3.73%	5.34%	20.35 %

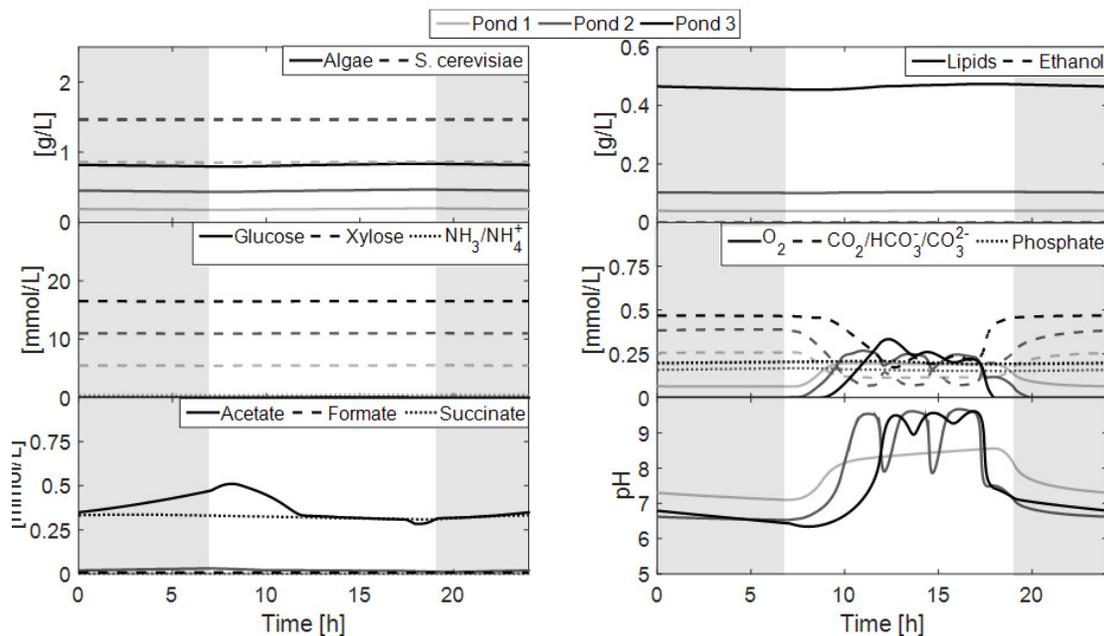


Figure 5-9: Concentrations of substrates, nutrients, and products in the three-pond system for Case 4. The dark areas of the plot represent nighttime. *S. cerevisiae* attains a higher biomass concentration than *C. reinhardtii*. Surprisingly, no ethanol is present in the system. This setup is unable to metabolize xylose, which accumulates in the system. The concentrations of oxygen and carbon dioxide change due to the photosynthetic activity of algae. Some acetate and succinate are accumulated in the last pond. The pH of the system stays between 5 and 9.

5.3 Conclusions

More ambitious environmental policies require better production processes for alternative forms of energy. Alternative liquid fuels are critical for the transportation sector as they have the high energy densities required for long distance travel. In this context, the efficient production of biodiesel from waste biomass or atmospheric CO₂ is critical.

Microbial consortia provide a way to transform both, lignocellulosic waste and atmospheric CO₂, into biodiesel. In this chapter, *C. reinhardtii* and *R. glutinis* are shown to work together to increase culture resilience, metabolize lignocellulosic sugars and capture CO₂ generated through respiration. This chapter also explores other heterotrophs such as *E. coli* and *S. cerevisiae*. In particular, it also shows that in most situations, only one heterotroph can occupy the heterotroph niche.

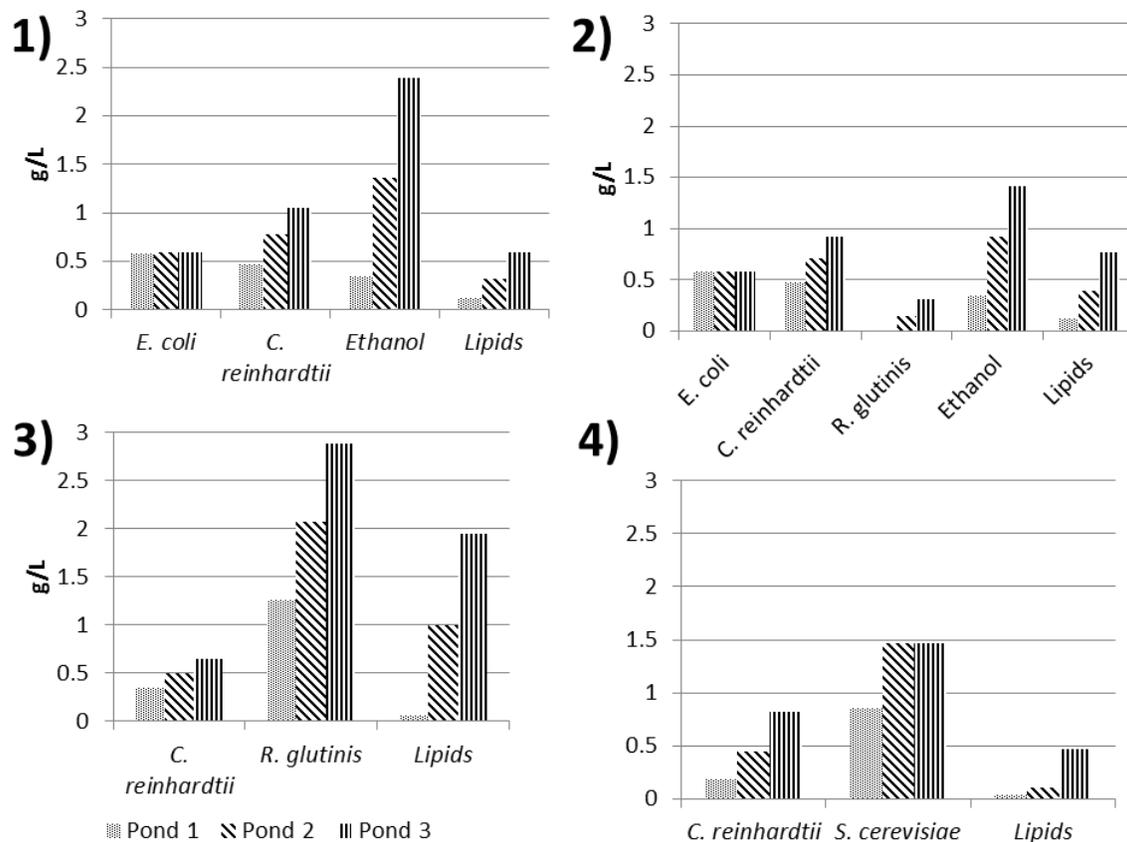


Figure 5-10: Concentration of biomass, ethanol, and lipids in each pond for each case. The case number is indicated at the top left corner of each plot.

The simulation framework utilized in this chapter enables the exploration of different process alternatives and setups. The simulations in this chapter predict that *C. reinhardtii* and *R. glutinis* cocultures can attain biodiesel prices of approximately \$1.41 / L, but this price could increase to up to \$ 3.54 if *E. coli* invades the cultivation system.

The results in this chapter are not optimized. With the availability of bioprocess models that are able to capture complex phenomena in raceway ponds, rigorous optimization will become possible in the near future. Sensitivities of DFBA systems, which are nondifferentiable, are challenging to compute. The work in Chapters 6, 7, and 8 enable the optimization of DFBA models. With optimization, parameter estimation and optimal design of raceway ponds for biofuels production will be possible.

Table 5.7: Economic analysis for biodiesel production.

	Case 1	Case 2	Case 3	Case 4	
<i>E. Coli</i> Production	11.2	11.1	0	0	gDW/(m ² × d)
<i>C. reinhardtii</i> Production	20.0	17.6	12.4	15.6	gDW/(m ² × d)
<i>S. cerevisiae</i> Production	0	0	0	27.9	gDW/(m ² × d)
<i>R. glutinis</i> Production	0	5.91	54.9	0	gDW/(m ² × d)
Lipids Production	40.8	53.2	135.2	32.3	tonne/(ha × yr)
Ethanol Production	166.2	98.1	0	0	tonne/(ha × yr)
Electricity AD	113.4	113.0	171	196.1	MWh/(ha × yr)
Capital Costs					
Raceway Pond	146	146	146	146	\$1000/ha
Non-pond Site Prep.	5	5	5	5	\$1000/ha
Engineering	22	22	22	22	\$1000/ha
Harvest	3.6	3.6	3.6	3.6	\$1000/ha
Extraction	4.6	4.6	4.6	4.6	\$1000/ha
Anaerobic Digester	16.5	16.4	24.8	28.5	\$1000/ha
TOTAL CAPITAL COSTS	197.6	197.5	206.0	209.6	\$1000/ha
Power Requirements					
Growth	7.8	7.8	7.8	7.8	MWh/(ha × yr)
Harvest	38	38	38	38	MWh/(ha × yr)
Dewatering	1.5	1.5	1.5	1.5	MWh/(ha × yr)
Electricity Required	-66.1	-65.7	-123.7	-148.8	MWh/(ha × yr)
Running Costs					
Labor	4430	4430	4430	4430	\$(ha × yr)
Electricity	-4440	-4410	-8320	-10000	\$(ha × yr)
Transesterification					
Power (Natural Gas)	188	245	623	149	\$(ha × yr)
Methanol	1870	2440	6200	1480	\$(ha × yr)
NaOH	877	739	808	3740	\$(ha × yr)
Water	145	145	145	145	\$(ha × yr)
Ammonia	1300	1560	2080	2300	\$(ha × yr)
DAP	2560	2560	2560	2190	\$(ha × yr)
Sugars	36200	36200	36200	36200	\$(ha × yr)
Anaerobic Digester	5690	5660	8570	9830	\$(ha × yr)
Capital Costs @ 10%/yr	19.8	19.8	20.6	21.0	\$1000/(ha × yr)
TOTAL COSTS	73.1	72.4	74.6	72.8	\$1000/(ha × yr)
Ethanol Profit	16.6	9.81	0	0	\$1000/(ha × yr)
Biodiesel Prod. Ideal	32.4	42.3	107.6	25.7	kL/(ha × yr)
Biodiesel Prod. Adjusted	16.2	21.2	53.8	12.9	kL/(ha × yr)
Biodiesel Cost	3.48	2.96	1.39	5.67	\$/L

Chapter 6

Sensitivities of Lexicographic Linear Programs

As explained in Chapter 3, a dynamic flux balance analysis (DFBA) model can be expressed as an ordinary differential equation (ODE) or differential-algebraic equation (DAE) system with lexicographic linear programs (LLP) embedded [56, 59]. In addition, LLPs are relevant in the context of goal-programming problems [63]. These types of models can also be embedded in optimization problems, equation-solving problems, or dynamic systems; therefore, the sensitivity analysis of LLPs is of interest. The computation of sensitivities for LLPs is challenging, as they are nonsmooth functions of their right-hand side. The work in this chapter shows how to compute sensitivities for LLPs and it is based on [47].

A LLP is a hierarchical fixed-priority linear program (LP). Equations (6.1) and (6.2) present a LLP parameterized by its right-hand side:

$$\begin{aligned} h_0(\mathbf{z}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_0^T \mathbf{v}, \\ &\text{s.t. } \mathbf{A}\mathbf{v} = \mathbf{z}, \\ &\quad \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{6.1}$$

and for $1 \leq i \leq n_h$

$$\begin{aligned}
 h_i(\mathbf{z}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_i^T \mathbf{v}, \\
 \text{s.t. } & \begin{bmatrix} \mathbf{A} \\ \mathbf{c}_0^T \\ \vdots \\ \mathbf{c}_{i-1}^T \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{z} \\ h_0(\mathbf{z}) \\ \vdots \\ h_{i-1}(\mathbf{z}) \end{bmatrix}, \\
 & \mathbf{v} \geq \mathbf{0}.
 \end{aligned} \tag{6.2}$$

A more compact way of writing LPs (6.1) and (6.2) is:

$$\begin{aligned}
 \mathbf{h}(\mathbf{z}) &= \text{lex} \min_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{C}^T \mathbf{v}, \\
 \text{s.t. } & \mathbf{A} \mathbf{v} = \mathbf{z}, \\
 & \mathbf{v} \geq \mathbf{0},
 \end{aligned} \tag{6.3}$$

LLPs can be applied in biochemical processes, economics, among many other topics. For example, DFBA can be used to model the dynamics of bioreactors. The optimization of DFBA systems will enable the optimal design of industrial bioprocesses employing microbial communities [60]. Another example comes from bilevel optimization, which is used to model Stackelberg games [138, 34], a popular model for markets in economics. When the lower-level optimization problem is a LP, this LP can be expressed as its equivalent KKT conditions to reformulate the bilevel problem as a single-level problem with equilibrium constraints. This reformulation results in bilinear terms which are difficult to handle by optimization algorithms due to violation of constraint qualifications and nonconvexity. In addition, sensitivity results for nonlinear programs typically require constraint qualifications that would be violated by equilibrium constraints. If all communication between the lower-level LP and upper-level optimization problem can be expressed as a sequence of hierarchical objective functions, a LLP is obtained. The optimal values function of a LLP is inherently more regular than the solution set of a LP as a function of its right-hand side [56].

Generalized derivative information would greatly increase the variety and efficiency of numerical methods that can be applied to problems with LLPs embedded.

In this chapter, we develop the mathematical theory required to compute elements of the B-subdifferential (limiting Jacobian) of LLPs as functions of their right-hand sides. The main tools employed here are Nesterov's lexicographic derivatives [96] and the notion of LD-derivatives [74]. The remainder of this chapter is structured as follows. First, we formally introduce the LLP, present basic theory about piecewise affine functions, and derive extensions of the chain rule for LD-derivatives. In Section 6.4, the theory for computing LD-derivatives of the optimal values of a LLP is developed. Section 6.5 presents examples illustrating the relevance of LLPs and the use of LD-derivatives. Finally, conclusions and future work are briefly discussed.

6.1 Definition of LLPs

Consider a LLP parameterized by its right-hand side. For each $i \in \{0, 1, \dots, n_h\}$, let $g^i : \mathbb{R}^{m+i} \rightarrow \overline{\mathbb{R}}$, where for $\mathbf{z} \in \mathbb{R}^m$:

$$\begin{aligned} g^0(\mathbf{z}) &= \inf_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_0^T \mathbf{v}, \\ \text{s.t. } \mathbf{A} \mathbf{v} &= \mathbf{z}, \\ \mathbf{v} &\geq \mathbf{0}, \end{aligned} \tag{6.4}$$

and for $1 \leq i \leq n_h$ and $\mathbf{z} \in \mathbb{R}^{m+i}$:

$$\begin{aligned} g^i(\mathbf{z}) &= \inf_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_i^T \mathbf{v}, \\ \text{s.t. } \begin{bmatrix} \mathbf{A} \\ \mathbf{c}_0^T \\ \vdots \\ \mathbf{c}_{i-1}^T \end{bmatrix} \mathbf{v} &= \mathbf{z}, \\ \mathbf{v} &\geq \mathbf{0}. \end{aligned} \tag{6.5}$$

Assumption 6.1.1. Let \mathbf{A} be of full row rank. For all $i \in \{0, \dots, n_h\}$, let \mathbf{A} and \mathbf{c}_j , $j = 0, \dots, i$, be such that $g^i(\mathbf{z}) > -\infty$ for all $\mathbf{z} \in \mathbb{R}^{m+i}$. In addition for all $i > 0$, let $\left[\mathbf{A}^T \quad \mathbf{c}_0 \quad \dots \quad \mathbf{c}_{i-1}\right]^T$ be full row rank.

Let $F = \{\mathbf{z} \in \mathbb{R}^m : -\infty < g^0(\mathbf{z}) < +\infty\}$, $\mathbf{A}^0 = \mathbf{A}$ and for $i \geq 1$ let $\mathbf{A}^i = \left[\mathbf{A}^T \quad \mathbf{c}_0 \quad \dots \quad \mathbf{c}_{i-1}\right]^T$. Under Assumption 6.1.1, \mathbf{h} in (6.1) and (6.2) is such that $\mathbf{h} : F \rightarrow \mathbb{R}^{n_h+1}$. In this chapter, only LLPs satisfying Assumption 6.1.1 will be considered.

Proposition 6.1.1. Let Assumption 6.1.1 hold, let $\mathbf{z} \in F$ and consider LPs (6.1) and (6.2). Then $-\infty < \mathbf{h}(\mathbf{z}) < +\infty$.

Proof: $-\infty < \mathbf{h}(\mathbf{z})$ by Assumption 6.1.1 and $h_0(\mathbf{z}) < +\infty$ because $\mathbf{z} \in F$. Therefore, there exists $\mathbf{v} \in \mathbb{R}^{n_v}$ such that $\mathbf{A}\mathbf{v} = \mathbf{z}$ and $\mathbf{c}_0^T \mathbf{v} = h_0(\mathbf{z}) < +\infty$. Now let $i \in \{1, \dots, n_h\}$. Assume that for all $j < i$, $h_j(\mathbf{z}) < +\infty$. This means there exists $\mathbf{v}_0 \geq \mathbf{0}$ such that $\mathbf{A}\mathbf{v}_0 = \mathbf{z}$ and for all $j < i$, $\mathbf{c}_j^T \mathbf{v}_0 = h_j(\mathbf{z})$. Therefore, $h_i(\mathbf{z}) \leq \mathbf{c}_i^T \mathbf{v}_0 < +\infty$. Since $h_0(\mathbf{z}) < +\infty$, the proof follows by induction. \square

Definition 6.1.1. Let Assumption 6.1.1 hold. Let $\mathbf{q}^i : F \rightarrow \mathbb{R}^{m+i}$ where $\mathbf{q}^0 : \mathbf{z} \mapsto \mathbf{z}$ and for $i = 1, \dots, n_h$, $\mathbf{q}^i : \mathbf{z} \mapsto \left[\mathbf{z}^T \quad h_0(\mathbf{z}) \quad \dots \quad h_{i-1}(\mathbf{z})\right]^T$. Let $\mathbf{h} : F \rightarrow \mathbb{R}^{n_h+1}$, which contains the objective function values in LPs (6.1) and (6.2) when $\mathbf{z} \in F$, and is such that $h_i = g^i \circ \mathbf{q}^i$.

Proposition 6.1.2. Consider LP (6.4) and suppose \mathbf{A} and \mathbf{c}_0 are such that $g^0(\mathbf{z}) > -\infty$ for all $\mathbf{z} \in \mathbb{R}^m$. \mathbf{A} is full row rank if and only if $\text{int}(F) \neq \emptyset$.

Proof: First we will show that if $\text{int}(F) \neq \emptyset$, then \mathbf{A} is full row rank. Assume \mathbf{A} is not full row rank and $\text{int}(F) \neq \emptyset$. Then, there exists at least one row that is linearly dependent on the other rows. Without loss of generality, let us assume that the last row of \mathbf{A} is linearly dependent. Let $\mathbf{b}_0 \in \text{int}(F)$. Then, any small perturbation of the last component of \mathbf{b}_0 while keeping all other components of \mathbf{b}_0 constant results in LP (6.4) becoming infeasible. Then, $\mathbf{b}_0 \notin \text{int}(F)$ and there is a contradiction.

For the next part of the proof assume \mathbf{A} is full row rank. Let $\mathbf{v} = \mathbf{1}$ and $\mathbf{b} = \mathbf{A}\mathbf{v}$. By hypothesis, this $\mathbf{b} \in F$. Let $\mathbf{d} \in \mathbb{R}^m$ and $\epsilon > 0$. Without loss of generality, let $\|\mathbf{d}\| = 1$. Let us find a $\Delta\mathbf{v}$ that satisfies, $\mathbf{A}(\mathbf{v} + \Delta\mathbf{v}) = \mathbf{b} + \epsilon\mathbf{d}$. Since $\mathbf{A}\mathbf{v} = \mathbf{b}$, we

need $\mathbf{A}\Delta\mathbf{v} = \epsilon\mathbf{d}$. Since \mathbf{A} is full row rank, the columns of \mathbf{A} span \mathbb{R}^m . Consider all possible collections of m columns that form a basis for the column space of \mathbf{A} . Take such a basis $\tilde{\mathbf{A}}$ and let $\Delta\tilde{\mathbf{v}} = \epsilon\tilde{\mathbf{A}}^{-1}\mathbf{d}$. Without loss of generality, let the columns of $\tilde{\mathbf{A}}$ be the first m columns of \mathbf{A} . Let $\Delta v_j = \Delta\tilde{v}_j$ if $j \in \{1, \dots, m\}$ and 0 otherwise. It is clear that $\Delta\mathbf{v}$ constructed in this form satisfies $\mathbf{A}(\mathbf{v} + \Delta\mathbf{v}) = \mathbf{b} + \epsilon\mathbf{d}$. Moreover, $\|\Delta\mathbf{v}\| = \|\Delta\tilde{\mathbf{v}}\|$. Then,

$$\|\Delta\mathbf{v}\| = \|\Delta\tilde{\mathbf{v}}\| \leq \epsilon\|\tilde{\mathbf{A}}^{-1}\|\|\mathbf{d}\| = \epsilon\|\tilde{\mathbf{A}}^{-1}\|,$$

and for small enough ϵ , $\|\Delta\mathbf{v}\| < 1$ which implies all components $|\Delta v_j| < 1$ and $\mathbf{v} + \Delta\mathbf{v} > \mathbf{0}$. Then, $\mathbf{b} + \epsilon\mathbf{d} \in F$ for small enough ϵ . By hypothesis $F = \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$ [12], so F is a nonempty convex set and $\mathbf{b} + \delta\mathbf{d} \in F$ for all $0 < \delta < \epsilon$. Since \mathbf{d} was arbitrary, $\mathbf{b} \in \text{int}(F)$ and $\text{int}(F) \neq \emptyset$. \square

Assumption 6.1.1 implies F is nonempty by Proposition 6.1.2. The necessity of \mathbf{A} being a full row rank matrix is not limiting in any sense. Linearly dependent rows imply some constraints are redundant. Deleting redundant constraints from \mathbf{A} and \mathbf{z} in LP (6.4) and redundant cost vectors in LP (6.5) results in LPs with the same optimal solution set and same objective function value.

For $i = \{0, \dots, n_h\}$, let $F_i = \{\mathbf{z} \in \mathbb{R}^{m+i} : -\infty < g^i(\mathbf{z}) < +\infty\}$. Under Assumption 6.1.1, all sets F_i are closed [56] and convex [12]. In addition, since F is nonempty by Proposition 6.1.2, then all sets F_i are nonempty by Proposition 6.1.1. Notice that $F = F_0$. The functions g^i are convex on the sets F_i [12].

Proposition 6.1.3. Let Assumption 6.1.1 hold and let $\mathbf{b} \in F$. Then, for all $i \geq 1$, all $\mathbf{q}^i(\mathbf{b}) \in F_i$.

Proof: By Assumption 6.1.1, $g^i(\mathbf{z}) > -\infty$ for any $\mathbf{z} \in \mathbb{R}^{m+i}, i \in \{0, \dots, n_h\}$. Then, we only need to show that $g^i(\mathbf{q}^i(\mathbf{b})) < +\infty$, which is equivalent to showing LP (6.5) is feasible, for $i = \{1, \dots, n_h\}$. Let $i \in \{1, \dots, n_h\}$ and assume $\mathbf{q}^{i-1}(\mathbf{b}) \in F_{i-1}$. Since $\mathbf{q}^{i-1}(\mathbf{b}) \in F_{i-1}$, there exists $\mathbf{v}_0 \geq \mathbf{0}$ such that $\mathbf{c}_{i-1}^T \mathbf{v}_0 = h_{i-1}(\mathbf{b})$ and $\mathbf{A}^{i-1} \mathbf{v}_0 = \mathbf{q}^{i-1}(\mathbf{b})$. Then $\mathbf{A}^i \mathbf{v}_0 = \mathbf{q}^i(\mathbf{b})$ and $g^i(\mathbf{q}^i(\mathbf{b})) = h_i(\mathbf{b}) \leq \mathbf{c}_i^T \mathbf{v}_0 < +\infty$. Therefore, $\mathbf{q}^i(\mathbf{b}) \in F_i$. Since $\mathbf{q}^0(\mathbf{b}) = \mathbf{b} \in F_0$, the proof follows by induction. \square

Proposition 6.1.3 implies that $\mathbf{q}^i : F \rightarrow F_i$ for all i .

6.2 Piecewise linear and piecewise affine functions

Definition 6.2.1. [119]. A continuous function $\boldsymbol{\rho} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is called *piecewise linear* (*affine*) if there exists a finite set of linear (affine) functions $\boldsymbol{\rho}_i : \mathbb{R}^m \rightarrow \mathbb{R}^n$ such that the inclusion $\boldsymbol{\rho}(\mathbf{x}) \in \{\boldsymbol{\rho}_1(\mathbf{x}), \dots, \boldsymbol{\rho}_k(\mathbf{x})\}$ holds for every $\mathbf{x} \in \mathbb{R}^m$. The linear (affine) functions $\boldsymbol{\rho}_i$ are called *selection functions*.

Since linear functions are also affine, piecewise linear functions are also piecewise affine. Every piecewise affine function is Lipschitz continuous (Proposition 2.2.7 in [119]). Piecewise affine functions are closed under composition [119]. In fact, given two piecewise affine functions $\boldsymbol{\rho} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $\boldsymbol{\zeta} : \mathbb{R}^n \rightarrow \mathbb{R}^p$, with affine selection functions $\{\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_k\}$ and $\{\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_l\}$, the selection functions of $\boldsymbol{\zeta} \circ \boldsymbol{\rho}$ are in the set that considers all possible compositions $\boldsymbol{\zeta}_i \circ \boldsymbol{\rho}_j$ for $i \in \{1, \dots, l\}$ and $j \in \{1, \dots, k\}$ [119]. Therefore, the composition of two piecewise linear functions is piecewise linear as all selection functions are linear functions and linear functions are closed under composition.

Lemma 6.2.1. [119]. Let $\boldsymbol{\rho} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ be a piecewise affine function. Then for any point $\mathbf{x}_0 \in \mathbb{R}^m$, there exists $\delta > 0$ such that for any \mathbf{x} such that $\|\mathbf{x} - \mathbf{x}_0\| < \delta$,

$$\boldsymbol{\rho}(\mathbf{x}) = \boldsymbol{\rho}(\mathbf{x}_0) + \boldsymbol{\rho}'(\mathbf{x}_0; \mathbf{x} - \mathbf{x}_0).$$

The proof of this lemma is in Section 2.2.2 in [119].

Proposition 6.2.2. Consider a piecewise affine function $\boldsymbol{\theta} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ and a function $\boldsymbol{\rho} : S \subset \mathbb{R}^m \rightarrow \mathbb{R}^n$. Suppose that $\boldsymbol{\rho}(\mathbf{x}) = \boldsymbol{\theta}(\mathbf{x})$ for all $\mathbf{x} \in S$. Given $\mathbf{d} \in \mathbb{R}^m$ and $\mathbf{x} \in S$ suppose that there exists $\delta_{\mathbf{d}}^*$ such that for $\tau \in (0, \delta_{\mathbf{d}}^*)$, $\mathbf{x} + \tau\mathbf{d} \in S$. Then, $\boldsymbol{\rho}$ has a directional derivative at \mathbf{x} in the direction \mathbf{d} and there exists $\delta_{\mathbf{d}} \in (0, \delta_{\mathbf{d}}^*]$ such that for $0 \leq \epsilon < \delta_{\mathbf{d}}$,

$$\boldsymbol{\rho}(\mathbf{x} + \epsilon\mathbf{d}) = \boldsymbol{\rho}(\mathbf{x}) + \boldsymbol{\rho}'(\mathbf{x}; \epsilon\mathbf{d}).$$

Proof: Notice that $\boldsymbol{\rho}(\mathbf{x} + \epsilon \mathbf{d}) = \boldsymbol{\theta}(\mathbf{x} + \epsilon \mathbf{d})$ holds $\forall \epsilon \in [0, \delta_{\mathbf{d}}^*]$. We know from Lemma 6.2.1 that there exists $\delta > 0$ such that:

$$\boldsymbol{\theta}(\mathbf{x} + \epsilon \mathbf{d}) = \boldsymbol{\theta}(\mathbf{x}) + \boldsymbol{\theta}'(\mathbf{x}; \epsilon \mathbf{d}), \forall \|\epsilon \mathbf{d}\| < \delta.$$

Let $\delta_{\mathbf{d}} = \min(\delta, \delta_{\mathbf{d}}^*)$. Since $\boldsymbol{\rho}(\mathbf{x} + \epsilon \mathbf{d}) = \boldsymbol{\theta}(\mathbf{x} + \epsilon \mathbf{d})$ for all $\epsilon \in (0, \delta_{\mathbf{d}})$, $\boldsymbol{\rho}(\mathbf{x} + \epsilon \mathbf{d}) = \boldsymbol{\rho}(\mathbf{x}) + \boldsymbol{\theta}'(\mathbf{x}; \epsilon \mathbf{d})$ for all $\epsilon \in (0, \delta_{\mathbf{d}})$. Now let us show that $\boldsymbol{\rho}'(\mathbf{x}; \epsilon \mathbf{d})$ exists and is equal to $\boldsymbol{\theta}'(\mathbf{x}; \epsilon \mathbf{d})$. Consider any sequence $\{\tau_i\}_{i=0}^{\infty}$, $\tau_i \in (0, \delta_{\mathbf{d}})$, $\tau_i \rightarrow 0$. From the definition of directional derivative:

$$\lim_{i \rightarrow \infty} \frac{\boldsymbol{\theta}(\mathbf{x} + \tau_i \mathbf{d}) - \boldsymbol{\theta}(\mathbf{x})}{\tau_i} = \boldsymbol{\theta}'(\mathbf{x}; \mathbf{d}).$$

Also, for all τ_i ,

$$\frac{\boldsymbol{\theta}(\mathbf{x} + \tau_i \mathbf{d}) - \boldsymbol{\theta}(\mathbf{x})}{\tau_i} = \frac{\boldsymbol{\rho}(\mathbf{x} + \tau_i \mathbf{d}) - \boldsymbol{\rho}(\mathbf{x})}{\tau_i}.$$

Therefore, $\boldsymbol{\rho}'(\mathbf{x}; \mathbf{d})$ exists and $\boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}) = \boldsymbol{\theta}'(\mathbf{x}; \mathbf{d})$. From Eqn. (2.3), $\boldsymbol{\rho}'(\mathbf{x}; \epsilon \mathbf{d}) = \boldsymbol{\theta}'(\mathbf{x}; \epsilon \mathbf{d})$ and $\boldsymbol{\rho}(\mathbf{x} + \epsilon \mathbf{d}) = \boldsymbol{\rho}(\mathbf{x}) + \boldsymbol{\rho}'(\mathbf{x}; \epsilon \mathbf{d})$ for all $\epsilon \in (0, \delta_{\mathbf{d}})$. \square

From now on we shall call functions piecewise linear or piecewise affine if they coincide with a piecewise linear or piecewise affine function on their domain of definition, even if their domain is not all \mathbb{R}^m .

Corollary 6.2.1. Let $\boldsymbol{\rho} : S \subset \mathbb{R}^m \rightarrow Z$ be a piecewise affine function and let $\mathbf{d} \in \mathbb{R}^m$ and $\mathbf{x} \in S$. If there exists $\delta_{\mathbf{d}}^* > 0$ such that for all $\tau \in (0, \delta_{\mathbf{d}}^*)$, $\mathbf{x} + \tau \mathbf{d} \in S$, then there exists $\delta_{\mathbf{d}}$ such that for all $\tau \in (0, \delta_{\mathbf{d}})$, $\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}) \in Z$.

Proof: From Proposition 6.2.2 there exists $\delta_{\mathbf{d}} > 0$ such that for $0 \leq \tau < \delta_{\mathbf{d}}$, $\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}) = \boldsymbol{\rho}(\mathbf{x}) + \boldsymbol{\rho}'(\mathbf{x}; \tau \mathbf{d}) = \boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d})$ (Eqn. 2.3). Since $\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}) \in Z$, then $\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}) \in Z$ for all $\tau \in (0, \delta_{\mathbf{d}})$.

Proposition 6.2.3. Let $X \subset \mathbb{R}^m$ be open, $\boldsymbol{\rho} : X \rightarrow \mathbb{R}^n$ be PC^1 , $\mathbf{x} \in X$, and $\mathbf{M} \in \mathbb{R}^{m \times q}$. Then $\boldsymbol{\rho}$ is l -smooth at \mathbf{x} and the sequence of functions (2.2) are all

piecewise linear.

Proof: $\boldsymbol{\rho}$ is l -smooth at \mathbf{x} because it is piecewise differentiable [74]. Let $\{\boldsymbol{\rho}^1, \dots, \boldsymbol{\rho}^k\}$ be a set of selection functions for $\boldsymbol{\rho}$ at \mathbf{x} with $k \in \mathbb{N}$. From Proposition 4.1.3 in [119] for $i \in \{1, \dots, n\}$, $\mathbf{d} \in \mathbb{R}^m$, $\rho'_i(\mathbf{x}; \mathbf{d}) \in \{\nabla \rho_i^1(\mathbf{x})^\top \mathbf{d}, \dots, \nabla \rho_i^k(\mathbf{x})^\top \mathbf{d}\}$. From Theorem 3.1.2 in [119], $\boldsymbol{\rho}'(\mathbf{x}; \cdot)$ is globally Lipschitz continuous. It is clear then that all components of the function $\boldsymbol{\rho}'(\mathbf{x}; \cdot)$ are piecewise linear, and therefore $\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(0)}$ is piecewise linear and PC^1 . It follows that for $j \in \{1, \dots, q\}$, $\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j)}$ are piecewise linear. \square

6.3 Extensions of directional derivatives

First, let us introduce the concept of directional derivative used in this chapter. Notice that unlike the standard definition presented in Definition 2.1.2, the set X is not required to be open.

Definition 6.3.1. Let $X \subset \mathbb{R}^m$ and $\boldsymbol{\rho} : X \rightarrow \mathbb{R}^n$. Let $\mathbf{x} \in X$ and $\mathbf{d} \in \mathbb{R}^m$ be such that there exists $\delta_{\mathbf{d}} > 0$ such that for all $\epsilon \in (0, \delta_{\mathbf{d}})$, $\mathbf{x} + \epsilon \mathbf{d} \in X$. The (one-sided) *directional derivative* of $\boldsymbol{\rho}$ at $\mathbf{x} \in X$ in the direction $\mathbf{d} \in \mathbb{R}^m$ is given by the following limit if it exists:

$$\boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}) \equiv \lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}) - \boldsymbol{\rho}(\mathbf{x})}{\tau}.$$

If at \mathbf{x} , the limit exists in \mathbb{R}^n for all directions $\mathbf{d} \in \mathbb{R}^m$, then, $\boldsymbol{\rho}$ is said to be *directionally differentiable* at \mathbf{x} .

If $\mathbf{x} \in \text{int}(X)$, there exists $\delta_{\mathbf{d}} > 0$ such that for all $\epsilon \in (0, \delta_{\mathbf{d}})$, $\mathbf{x} + \epsilon \mathbf{d} \in X$ for any $\mathbf{d} \in \mathbb{R}^m$. Therefore, if $\mathbf{x} \in \text{int}(X)$, Definition 6.3.1 reduces to the standard definition of directional derivative.

Assumption 6.3.1. Let $X \subset \mathbb{R}^m$ be open, $Z \subset \mathbb{R}^n$, $\boldsymbol{\rho} : X \rightarrow Z$, $\boldsymbol{\zeta} : Z \rightarrow \mathbb{R}^p$ and $\boldsymbol{\sigma} : X \rightarrow \mathbb{R}^p$, where $\boldsymbol{\sigma} \equiv \boldsymbol{\zeta} \circ \boldsymbol{\rho}$. Assume $\boldsymbol{\rho}$ is l -smooth at $\mathbf{x} \in X$ and $\boldsymbol{\zeta}$ is locally Lipschitz continuous. Note that Z is not required to be open.

Definition 6.3.2. Let Assumption 6.3.1 hold, let $\mathbf{K} \in \mathbb{R}^{m \times q}$, $\mathbf{M} = \boldsymbol{\rho}'(\mathbf{x}; \mathbf{K})$ and consider the sets $G_{\mathbf{x}}^{(i)} = \{\boldsymbol{\rho}_{\mathbf{x}, \mathbf{Z}}^{(i)}(\mathbf{z}_{i+1}) : \mathbf{Z} \in \mathbb{R}^{m \times q}, \mathbf{z}_{q+1} \in \mathbb{R}^m\} \subset \mathbb{R}^n$ for $i \in \{0, \dots, q\}$. ζ is said to be $\boldsymbol{\rho}$ -weakly l -smooth at \mathbf{x} if for any matrix \mathbf{K} the following sequence is well-defined:

$$\begin{aligned} \zeta_{\boldsymbol{\rho}(\mathbf{x}), \mathbf{M}}^{(0)} : G_{\mathbf{x}}^{(0)} &\rightarrow \mathbb{R}^p : \mathbf{d} \mapsto \zeta'(\boldsymbol{\rho}(\mathbf{x}); \mathbf{d}), \\ \zeta_{\boldsymbol{\rho}(\mathbf{x}), \mathbf{M}}^{(j)} : G_{\mathbf{x}}^{(j)} &\rightarrow \mathbb{R}^p : \mathbf{d} \mapsto [\zeta_{\boldsymbol{\rho}(\mathbf{x}), \mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \mathbf{d}), \quad \forall j \in \{1, \dots, q\}. \end{aligned} \quad (6.6)$$

Remark 6.3.1. Let Assumption 6.3.1 hold, let $\mathbf{K} \in \mathbb{R}^{m \times q}$, $\mathbf{M} = \boldsymbol{\rho}'(\mathbf{x}; \mathbf{K})$, and let ζ be l -smooth at $\boldsymbol{\rho}(\mathbf{x})$. Then, ζ is $\boldsymbol{\rho}$ -weakly l -smooth at \mathbf{x} .

Proposition 6.3.1. Let $Z \in \mathbb{R}^n$ and $\zeta : Z \rightarrow \mathbb{R}^p$ be Lipschitz near $\mathbf{z} \in Z$. Let the set $G_{\mathbf{z}}$ contain $\mathbf{d} \in \mathbb{R}^n$ such that $\zeta'(\mathbf{z}; \mathbf{d})$ exists. Then $\zeta'(\mathbf{z}; \cdot)$ is globally Lipschitz on $G_{\mathbf{z}}$.

Proof: The proof is basically the same as the one presented in Theorem 3.1.2 in [119] only that the assumption of directional differentiability is dropped. Let $\mathbf{d}_1, \mathbf{d}_2 \in G_{\mathbf{z}}$. Then,

$$\begin{aligned} \|\zeta'(\mathbf{z}; \mathbf{d}_1) - \zeta'(\mathbf{z}; \mathbf{d}_2)\| &= \lim_{\tau \rightarrow 0^+} \frac{\|\zeta(\mathbf{z} + \tau \mathbf{d}_1) - \zeta(\mathbf{z} + \tau \mathbf{d}_2)\|}{\tau} \\ &\leq \lim_{\tau \rightarrow 0^+} L \frac{\|\tau \mathbf{d}_1 - \tau \mathbf{d}_2\|}{\tau} = L \|\mathbf{d}_1 - \mathbf{d}_2\|, \end{aligned}$$

where L is the Lipschitz constant for ζ near \mathbf{z} . Therefore, $\zeta'(\mathbf{z}; \cdot)$ is globally Lipschitz continuous. \square

Corollary 6.3.1. Let Assumption 6.3.1 hold and let ζ be $\boldsymbol{\rho}$ -weakly l -smooth at \mathbf{x} . Then the functions in the sequence (6.6) are globally Lipschitz continuous.

Proof: Let us assume that for $j \in \{1, \dots, q\}$, $\zeta_{\boldsymbol{\rho}(\mathbf{x}), \mathbf{M}}^{(j-1)}$ is globally Lipschitz continuous on $G_{\mathbf{x}}^{(j-1)}$ with constant L . By Proposition 6.3.1, $\zeta_{\boldsymbol{\rho}(\mathbf{x}), \mathbf{M}}^{(j)}$ is globally Lipschitz continuous. Since $\zeta_{\boldsymbol{\rho}(\mathbf{x}), \mathbf{M}}^{(0)}$ is globally Lipschitz continuous from Proposition 6.3.1, the proof follows by induction. \square

The following Theorem is a generalization of the chain rule for l -smooth functions.

Theorem 6.3.2. Let Assumption 6.3.1 hold. Suppose that ζ is ρ -weakly l -smooth at \mathbf{x} . Then σ is l -smooth at \mathbf{x} and for $\mathbf{M} \in \mathbb{R}^{m \times q}$, $\sigma'(\mathbf{x}; \mathbf{M}) = \zeta'(\rho(\mathbf{x}); \rho'(\mathbf{x}; \mathbf{M}))$.

Proof: For ρ to be l -smooth, it is necessarily Lipschitz continuous near \mathbf{x} . Since ζ is locally Lipschitz continuous, the composition $\sigma \equiv \zeta \circ \rho$ is Lipschitz continuous near \mathbf{x} . First, we will show that σ is directionally differentiable at \mathbf{x} .

We need to show that the following limit exists in \mathbb{R}^p for all $\mathbf{d} \in \mathbb{R}^m$:

$$\lim_{\tau \rightarrow 0^+} \frac{\sigma(\mathbf{x} + \tau \mathbf{d}) - \sigma(\mathbf{x})}{\tau} = \lim_{\tau \rightarrow 0^+} \frac{\zeta(\rho(\mathbf{x} + \tau \mathbf{d})) - \zeta(\rho(\mathbf{x}))}{\tau}.$$

Given $\mathbf{d} \in \mathbb{R}^m$, by assumption, we know the following limit exists in \mathbb{R}^p :

$$\zeta'(\rho(\mathbf{x}); \rho'(\mathbf{x}; \mathbf{d})) = \lim_{\tau \rightarrow 0^+} \frac{\zeta(\rho(\mathbf{x}) + \tau \rho'(\mathbf{x}; \mathbf{d})) - \zeta(\rho(\mathbf{x}))}{\tau}.$$

We also know from the definition of the directional derivative that:

$$\lim_{\tau \rightarrow 0^+} \frac{\rho(\mathbf{x} + \tau \mathbf{d}) - \rho(\mathbf{x}) - \tau \rho'(\mathbf{x}; \mathbf{d})}{\tau} = \mathbf{0}.$$

We will show that

$$\lim_{\tau \rightarrow 0^+} \frac{\zeta(\rho(\mathbf{x}) + \tau \rho'(\mathbf{x}; \mathbf{d})) - \zeta(\rho(\mathbf{x}))}{\tau} = \lim_{\tau \rightarrow 0^+} \frac{\zeta(\rho(\mathbf{x} + \tau \mathbf{d})) - \zeta(\rho(\mathbf{x}))}{\tau},$$

which is equivalent to showing that

$$\lim_{\tau \rightarrow 0^+} \frac{\zeta(\rho(\mathbf{x}) + \tau \rho'(\mathbf{x}; \mathbf{d})) - \zeta(\rho(\mathbf{x} + \tau \mathbf{d}))}{\tau} = \mathbf{0}.$$

From the existence of the directional derivative $\zeta'(\rho(\mathbf{x}); \rho'(\mathbf{x}; \mathbf{d}))$ there exists $\epsilon_1 > 0$:

$$\rho(\mathbf{x}) + \tau \rho'(\mathbf{x}; \mathbf{d}) \in Z, \forall \tau \in (0, \epsilon_1).$$

Since ζ is locally Lipschitz continuous there exists $\epsilon_2 > 0$ and $K > 0$:

$$\begin{aligned} \|\zeta(\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d})) - \zeta(\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))\| &\leq K \|\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}) - (\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))\|, \\ \forall \tau \in (0, \epsilon_1) : \|\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}) - (\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))\| &< \epsilon_2, \end{aligned}$$

which is equivalent to

$$\begin{aligned} \left\| \frac{\zeta(\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d})) - \zeta(\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))}{\tau} \right\| &\leq K \left\| \frac{\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}) - (\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))}{\tau} \right\|, \\ \forall \tau \in (0, \epsilon_1) : \|\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}) - (\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))\| &< \epsilon_2. \end{aligned}$$

Since, by the existence of the directional derivative $\boldsymbol{\rho}'(\mathbf{x}; \mathbf{d})$,

$$\lim_{\tau \rightarrow 0^+} \left\| \frac{\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}) - (\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))}{\tau} \right\| = 0,$$

it follows that

$$\lim_{\tau \rightarrow 0^+} \left\| \frac{\zeta(\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d})) - \zeta(\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))}{\tau} \right\| = 0.$$

Since $\mathbf{d} \in \mathbb{R}^m$ was arbitrary, $\boldsymbol{\sigma}$ is directionally differentiable at \mathbf{x} and

$$\boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(0)}(\mathbf{d}) = \zeta_{\boldsymbol{\rho}(\mathbf{x}), \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M})}^{(0)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(0)}(\mathbf{d})).$$

Now let us assume that for $j \in \{1, \dots, q\}$, for all $\mathbf{M} \in \mathbb{R}^{m \times q}$ and $\mathbf{m}_{q+1} \in \mathbb{R}^m$,

$$\boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) = \zeta_{\boldsymbol{\rho}(\mathbf{x}), \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M})}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j)).$$

For brevity, let $\mathbf{y} \equiv \boldsymbol{\rho}(\mathbf{x})$ and $\mathbf{Y} \equiv \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M})$. Also, let $\hat{\mathbf{m}} \equiv \mathbf{m}_{j+1}$. We need to show

that the following limit exists in \mathbb{R}^p for all $\mathbf{M} \in \mathbb{R}^{m \times q}$ and $\mathbf{m}_{q+1} \in \mathbb{R}^m$:

$$\begin{aligned} & \lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau \widehat{\mathbf{m}}) - \boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j)}{\tau} = \\ & \lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau \widehat{\mathbf{m}})) - \boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j))}{\tau}. \end{aligned}$$

Given $\mathbf{M} \in \mathbb{R}^{m \times q}$ and $\mathbf{m}_{q+1} \in \mathbb{R}^m$, since $\boldsymbol{\zeta}$ is $\boldsymbol{\rho}$ -weakly l -smooth at \mathbf{x} , we know the following limit exists in \mathbb{R}^p :

$$\begin{aligned} & [\boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}]'\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j), [\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'\left(\mathbf{m}_j; \widehat{\mathbf{m}}\right)\right) = \\ & \lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau [\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'\left(\mathbf{m}_j; \widehat{\mathbf{m}}\right)\right) - \boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j)\right)}{\tau}, \end{aligned}$$

and $\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau [\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'\left(\mathbf{m}_j; \widehat{\mathbf{m}}\right) \in G_{\mathbf{x}}^{(j-1)}$ for $\tau > 0$ small enough. We also know from the definition of l -smoothness that:

$$\lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau \widehat{\mathbf{m}}) - \boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) - \tau [\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'\left(\mathbf{m}_j; \widehat{\mathbf{m}}\right)}{\tau} = \mathbf{0}.$$

We will show that

$$\begin{aligned} & \lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau [\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'\left(\mathbf{m}_j; \widehat{\mathbf{m}}\right)\right) - \boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j)\right)}{\tau} = \\ & \lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau \widehat{\mathbf{m}})\right) - \boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j)\right)}{\tau}, \end{aligned}$$

which is equivalent to showing that

$$\lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau [\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'\left(\mathbf{m}_j; \widehat{\mathbf{m}}\right)\right) - \boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(j-1)}\left(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau \widehat{\mathbf{m}})\right)}{\tau} = \mathbf{0}.$$

From Corollary 6.3.1, $\boldsymbol{\zeta}_{\mathbf{y}, \mathbf{Y}}^{(k)}$ is globally Lipschitz continuous for all $k \in \{0, \dots, q\}$.

Then, there exists $\delta > 0$ such that $\tau \in [0, \delta)$ implies that $\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau [\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'\left(\mathbf{m}_j; \widehat{\mathbf{m}}\right) \in$

$G_{\mathbf{x}}^{(j-1)}$ and $K > 0$ such that

$$K \left\| \frac{\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau[\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}}) - \boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau\widehat{\mathbf{m}})}{\tau} \right\| \geq \left\| \frac{\boldsymbol{\zeta}_{\mathbf{y},\mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau[\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}})) - \boldsymbol{\zeta}_{\mathbf{y},\mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau\widehat{\mathbf{m}}))}{\tau} \right\|, \\ \forall \tau \in [0, \delta).$$

Since, by the existence of the directional derivative $[\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}})$,

$$\lim_{\tau \rightarrow 0^+} \left\| \frac{\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau[\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}}) - \boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau\widehat{\mathbf{m}})}{\tau} \right\| = 0,$$

it follows that

$$\lim_{\tau \rightarrow 0^+} \left\| \frac{\boldsymbol{\zeta}_{\mathbf{y},\mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau[\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}})) - \boldsymbol{\zeta}_{\mathbf{y},\mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau\widehat{\mathbf{m}}))}{\tau} \right\| = 0.$$

Since $\mathbf{M} \in \mathbb{R}^{m \times q}$ and $\mathbf{m}_{q+1} \in \mathbb{R}^m$ were arbitrary, $\boldsymbol{\sigma}_{\mathbf{x},\mathbf{M}}^{(j-1)}$ is directionally differentiable at \mathbf{m}_j and for all $\mathbf{M} \in \mathbb{R}^{m \times q}$ and $\mathbf{m}_{q+1} \in \mathbb{R}^m$

$$\boldsymbol{\sigma}_{\mathbf{x},\mathbf{M}}^{(j)}(\widehat{\mathbf{m}}) = \boldsymbol{\zeta}_{\mathbf{y},\mathbf{Y}}^{(j)}(\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j)}(\widehat{\mathbf{m}})).$$

Since $\boldsymbol{\sigma}_{\mathbf{x},\mathbf{M}}^{(0)}(\mathbf{m}_1) = \boldsymbol{\zeta}_{\mathbf{y},\mathbf{Y}}^{(0)}(\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(0)}(\mathbf{m}_1))$ for any $\mathbf{M} \in \mathbb{R}^{m \times q}$, the rest of the LD-derivatives follow by induction. \square

Remark 6.3.2. Notice that Theorem 6.3.2 provides a chain rule when $\boldsymbol{\rho}(\mathbf{x}) \in \text{bnd}(Z)$ which is not possible in the classical theory.

Theorem 6.3.3. Let Assumption 6.3.1 hold. Suppose that $\boldsymbol{\sigma}$ is l -smooth at \mathbf{x} . Let $\boldsymbol{\rho}(\mathbf{x}) \in \text{bnd}(Z)$. Assume that for all $\mathbf{d} \in \mathbb{R}^m$, there exists $\delta_{\mathbf{d}} > 0$ such that for any $\tau \in (0, \delta_{\mathbf{d}})$, $\boldsymbol{\rho}(\mathbf{x}) + \tau\boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}) \in Z$. Let $G_{\mathbf{x}}^{(j)}$ be the sets described in Definition 6.3.2. Assume that for any $\mathbf{m}_{q+1} \in \mathbb{R}^m$, $\mathbf{M} \in \mathbb{R}^{m \times q}$ and $j \in \{0, \dots, q-1\}$, there exists $\delta_{\mathbf{m}_{j+2}} > 0$ such that for all $\tau \in (0, \delta_{\mathbf{m}_{j+2}})$, $\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j)}(\mathbf{m}_{j+1}) + \tau[\boldsymbol{\rho}_{\mathbf{x},\mathbf{M}}^{(j)}]'(\mathbf{m}_{j+1}; \mathbf{m}_{j+2}) \in G_{\mathbf{x}}^{(j)}$. Then, $\boldsymbol{\zeta}$ is $\boldsymbol{\rho}$ -weakly l -smooth at \mathbf{x} and for $\mathbf{M} \in \mathbb{R}^{m \times q}$, $\boldsymbol{\sigma}'(\mathbf{x}; \mathbf{M}) = \boldsymbol{\zeta}'(\boldsymbol{\rho}(\mathbf{x}); \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M}))$.

Proof: First we show the limit for the directional derivative. Given $\mathbf{d} \in \mathbb{R}^m$ we want to show that the following limit exists in \mathbb{R}^p :

$$\zeta'(\boldsymbol{\rho}(\mathbf{x}); \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d})) = \lim_{\tau \rightarrow 0^+} \frac{\zeta(\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d})) - \zeta(\boldsymbol{\rho}(\mathbf{x}))}{\tau}.$$

We know that the following limit exists in \mathbb{R}^p :

$$\lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\sigma}(\mathbf{x} + \tau \mathbf{d}) - \boldsymbol{\sigma}(\mathbf{x})}{\tau} = \lim_{\tau \rightarrow 0^+} \frac{\zeta(\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d})) - \zeta(\boldsymbol{\rho}(\mathbf{x}))}{\tau}.$$

If we show that

$$\lim_{\tau \rightarrow 0^+} \frac{\zeta(\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d})) - \zeta(\boldsymbol{\rho}(\mathbf{x}))}{\tau} = \lim_{\tau \rightarrow 0^+} \frac{\zeta(\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d})) - \zeta(\boldsymbol{\rho}(\mathbf{x}))}{\tau},$$

which is equivalent to showing that

$$\lim_{\tau \rightarrow 0^+} \frac{\zeta(\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d})) - \zeta(\boldsymbol{\rho}(\mathbf{x} + \tau \mathbf{d}))}{\tau} = \mathbf{0}, \quad (6.7)$$

the proof is complete. Notice that $\zeta(\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}))$ is well defined by the assumption of $\boldsymbol{\rho}(\mathbf{x}) + \tau \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d}) \in Z$ for all $\tau \in (0, \delta_{\mathbf{d}})$. Since ζ is locally Lipschitz, the proof of Theorem 6.3.2 establishes (6.7). Furthermore, $\boldsymbol{\sigma}'(\mathbf{x}; \mathbf{d}) = \zeta'(\boldsymbol{\rho}(\mathbf{x}); \boldsymbol{\rho}'(\mathbf{x}; \mathbf{d})) = \boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(0)}(\mathbf{d}) = \zeta_{\boldsymbol{\rho}(\mathbf{x}), \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M})}^{(0)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(0)}(\mathbf{d}))$. Since $\mathbf{d} \in \mathbb{R}^m$ was arbitrary, the directional derivative of ζ at $\boldsymbol{\rho}(\mathbf{x})$ exists in all directions $\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(0)}(\mathbf{d})$ and the first function of sequence (6.6) is well-defined with its corresponding domain $G_{\mathbf{x}}^{(0)} = \{\boldsymbol{\rho}_{\mathbf{x}, \mathbf{Z}}^{(0)}(\mathbf{z}_1) : \mathbf{Z} \in \mathbb{R}^{m \times q}, \mathbf{z}_{q+1} \in \mathbb{R}^m\}$. By Proposition 6.3.1, $\zeta_{\boldsymbol{\rho}(\mathbf{x}), \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M})}^{(0)}$ is globally Lipschitz on $G_{\mathbf{x}}^{(0)}$.

Now let us assume that for $j \in \{1, \dots, q\}$ and for all $\mathbf{M} \in \mathbb{R}^{m \times q}$ and $\mathbf{m}_{j+1} \in \mathbb{R}^m$,

$$\boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) = \zeta_{\boldsymbol{\rho}(\mathbf{x}), \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M})}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j)),$$

and that the function $\zeta_{\boldsymbol{\rho}(\mathbf{x}), \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M})}^{(j-1)}$ is well-defined in the sense of (6.6) with $G_{\mathbf{x}}^{(j-1)} = \{\boldsymbol{\rho}_{\mathbf{x}, \mathbf{Z}}^{(j-1)}(\mathbf{z}_j) : \mathbf{Z} \in \mathbb{R}^{m \times q}, \mathbf{z}_{q+1} \in \mathbb{R}^m\}$ and globally Lipschitz on its domain. For brevity, let $\mathbf{y} \equiv \boldsymbol{\rho}(\mathbf{x})$ and $\mathbf{Y} \equiv \boldsymbol{\rho}'(\mathbf{x}; \mathbf{M})$. Also, let $\widehat{\mathbf{m}} \equiv \mathbf{m}_{j+1}$.

We want to show that the following limit exists in \mathbb{R}^p :

$$\begin{aligned} & [\zeta_{\mathbf{y}, \mathbf{Y}}^{(j-1)}]'(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j), [\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}})) = \\ & \lim_{\tau \rightarrow 0^+} \frac{\zeta_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau[\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}})) - \zeta_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j))}{\tau}. \end{aligned}$$

We know that the following limits exist in \mathbb{R}^p :

$$\begin{aligned} & \lim_{\tau \rightarrow 0^+} \frac{\boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau\widehat{\mathbf{m}}) - \boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j)}{\tau} = \\ & \lim_{\tau \rightarrow 0^+} \frac{\zeta_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau\widehat{\mathbf{m}})) - \zeta_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j))}{\tau}. \end{aligned}$$

Analogous to the proof in Theorem 6.3.2, if we show that

$$\lim_{\tau \rightarrow 0^+} \frac{\zeta_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau[\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}})) - \zeta_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \tau\widehat{\mathbf{m}}))}{\tau} = \mathbf{0}, \quad (6.8)$$

the proof is complete. By assumption, the quantity $\zeta_{\mathbf{y}, \mathbf{Y}}^{(j-1)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) + \tau[\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j-1)}]'(\mathbf{m}_j; \widehat{\mathbf{m}}))$ is well-determined for $\tau > 0$ small enough, $\mathbf{m}_{q+1} \in \mathbb{R}^m$, $\mathbf{M} \in \mathbb{R}^{m \times q}$. Then, the proof of Theorem 6.3.2 establishes (6.8) and

$$\boldsymbol{\sigma}_{\mathbf{x}, \mathbf{M}}^{(j)}(\widehat{\mathbf{m}}) = \zeta_{\mathbf{y}, \mathbf{Y}}^{(j)}(\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j)}(\widehat{\mathbf{m}})).$$

Since \mathbf{M} and \mathbf{m}_{q+1} are arbitrary, $\zeta_{\mathbf{y}, \mathbf{Y}}^{(j)}$ has as its domain $G_{\mathbf{x}}^{(j)} = \{\boldsymbol{\rho}_{\mathbf{x}, \mathbf{Z}}^{(j)}(\mathbf{z}_{j+1}) : \mathbf{Z} \in \mathbb{R}^{m \times q}, \mathbf{z}_{j+1} \in \mathbb{R}^m\}$ and by Proposition 6.3.1 $\zeta_{\mathbf{y}, \mathbf{Y}}^{(j)}$ is globally Lipschitz on its domain. Since the case for $j = 0$ was established, the proof follows by induction. \square

Remark 6.3.3. The assumption that for any $\mathbf{m}_{q+1} \in \mathbb{R}^m$, $\mathbf{M} \in \mathbb{R}^{m \times q}$ and $j \in \{0, \dots, q-1\}$, there exists $\delta_{\mathbf{m}_{j+2}} > 0$ such that for all $\tau \in (0, \delta_{\mathbf{m}_{j+2}})$, $\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1}) + \tau[\boldsymbol{\rho}_{\mathbf{x}, \mathbf{M}}^{(j)}]'(\mathbf{m}_{j+1}; \mathbf{m}_{j+2}) \in G_{\mathbf{x}}^{(j)}$ may seem difficult to verify at first glance. However, if Assumption 6.3.1 holds and in addition $\boldsymbol{\rho}$ is a PC^1 function, then this assumption follows from Proposition 6.2.3 and Corollary 6.2.1.

Theorems 6.3.2 and 6.3.3 are extensions of Theorem 3.1.1 in [119] and the chain

rule (2.5) from [96] under weaker assumptions. In these propositions, ζ is not required to be directionally differentiable; its directional derivatives need to exist in \mathbb{R}^p only in certain directions. The chain rule in [27] cannot be applied if $\rho(\mathbf{x}) \in \text{bnd}(Z)$; however, Theorems 6.3.2 and 6.3.3 can deal with this situation.

6.4 LD-derivatives of lexicographic linear programs

Before computing the LD-derivatives of LLPs, we discuss why results in the literature such as Proposition 4.12 in [14] are not applicable. This proposition refers to optimization problems of the form

$$\min_{\mathbf{x} \in X} f(\mathbf{x}, \mathbf{u}) \text{ subject to } \mathbf{x} \in \Phi, \quad (6.9)$$

where $\mathbf{u} \in U$ is a parameter vector and Φ is nonempty and closed. This proposition also requires the following definition.

Definition 6.4.1. Consider the optimization problem (6.9). The inf-compactness condition holds at $\mathbf{u}_0 \in U$ if there exists $\alpha \in \mathbb{R}$ and a compact set $C \subset X$ such that for every \mathbf{u} near \mathbf{u}_0 , the level set $\text{lev}_\alpha f(\cdot, \mathbf{u}) := \{\mathbf{x} \in \Phi : f(\mathbf{x}, \mathbf{u}) \leq \alpha\}$ is nonempty and contained in C .

Next we present Proposition 4.12 in [14]:

Proposition 6.4.1. Suppose that

1. the function $f(\mathbf{x}, \mathbf{u})$ is continuous on $X \times U$,
2. the inf-compactness condition holds at \mathbf{u}_0 ,
3. for any $\mathbf{x} \in \Phi$ the function $f_{\mathbf{x}}(\cdot) := f(\mathbf{x}, \cdot)$ is directionally differentiable at \mathbf{u}_0 ,
4. if $\mathbf{d} \in U$, $t_n \downarrow 0$ and $\{\mathbf{x}_n\}$ is a sequence in C , then $\{\mathbf{x}_n\}$ has a limit point $\bar{\mathbf{x}}$ such that

$$\limsup_{n \rightarrow \infty} \frac{f(\mathbf{x}_n, \mathbf{u}_0 + t_n \mathbf{d}) - f(\mathbf{x}_n, \mathbf{u}_0)}{t_n} \geq f'_{\bar{\mathbf{x}}}(\mathbf{u}_0; \mathbf{d}). \quad (6.10)$$

Then the optimal value function $v(\mathbf{u})$ is directionally differentiable at \mathbf{u}_0 and

$$v'(\mathbf{u}_0; \mathbf{d}) = \inf_{\mathbf{x} \in S(\mathbf{u}_0)} f'_{\mathbf{x}}(\mathbf{u}_0; \mathbf{d}), \quad (6.11)$$

where $S(\mathbf{u}) := \arg \min_{\mathbf{x} \in \Phi} f(\mathbf{x}, \mathbf{u})$. Moreover, if $\mathbf{x}_n \in S(\mathbf{u}_0 + t_n \mathbf{d})$ for some $t_n \downarrow 0$, then any limit point $\bar{\mathbf{x}}$ of $\{\mathbf{x}_n\}$ belongs to $S_1(\mathbf{u}_0, \mathbf{d})$, where $S_1(\mathbf{u}_0, \mathbf{d}) := \arg \min_{\mathbf{x} \in S(\mathbf{u}_0)} f'_{\mathbf{x}}(\mathbf{u}_0, \mathbf{d})$.

To apply Proposition 6.4.1, we need to consider the duals of each LP in (6.1) and (6.2):

$$\begin{aligned} h_i(\mathbf{z}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^{m+i}} [\mathbf{q}^i(\mathbf{z})]^T \boldsymbol{\lambda} \\ &\text{s.t. } [\mathbf{A}^i]^T \boldsymbol{\lambda} \leq \mathbf{c}_i. \end{aligned} \quad (6.12)$$

For $i \in \{0, \dots, n_h\}$, let $\mathcal{D}^i \subset \mathbb{R}^{m+i}$ be the feasible set of the dual of LP (6.12). Since Proposition 6.4.1 considers a minimization problem, $f^i(\boldsymbol{\lambda}, \mathbf{z}) = -[\mathbf{q}^i(\mathbf{z})]^T \boldsymbol{\lambda}$. Notice that for $i \in \{0, \dots, n_h\}$, the feasible set of (6.12) is independent of \mathbf{z} and nonempty under Assumption 6.1.1. The next propositions show that the inf-compactness condition cannot be satisfied by LLPs.

Proposition 6.4.2. Let Assumption 6.1.1 hold and consider LP (6.12). Then for all i , the inf-compactness condition is not satisfied at \mathbf{z}_0 if $\mathbf{q}^i(\mathbf{z}_0) \in \text{bnd}(F_i)$.

Proof: F_i is closed [56]. Let us assume $\mathbf{q}^i(\mathbf{z}_0) \in \text{bnd}(F_i)$. Since $\mathbf{q}^i(\mathbf{z}_0) \in F_i$, the solution set of LP (6.12) is nonempty, and it is closed and convex [12]. Therefore, by Proposition 3.5 in [60] the solution set of LP (6.12) is unbounded. Since the optimal level set is unbounded and this is the smallest nonempty level set, there is no nonempty bounded level set at \mathbf{z}_0 and the inf-compactness condition is not satisfied. \square

Proposition 6.4.3. Let Assumption 6.1.1 hold. Then for all $i > 0$, $\mathbf{q}^i(F) \subset \text{bnd}(F_i)$.

Proof: By Proposition 6.1.3, $\mathbf{q}^i(F) \subset F_i$. In addition, we know that $q_{m+i}^i(\mathbf{z}) = h_{i-1}(\mathbf{z})$. Let $\mathbf{d} = -\mathbf{e}_{m+i}$. Then for any $\epsilon > 0$, $\mathbf{q}^i(\mathbf{z}) + \epsilon \mathbf{d} \notin F_i$ because the $(m+i)^{\text{th}}$ component of $\mathbf{q}^i(\mathbf{z})$ is $h_{i-1}(\mathbf{z})$ which is optimal and therefore, any value less

than $h_{i-1}(\mathbf{z})$ results in an infeasible LP. Then $\mathbf{q}^i(\mathbf{z}) \in \text{bnd}(F_i)$ and thus $\mathbf{q}^i(F) \subset \text{bnd}(F_i)$. \square

Therefore by Propositions 6.4.2 and 6.4.3, Proposition 6.4.1 cannot be applied to LLPs.

6.4.1 Computation of LD-derivatives of LLPs

In this section, we derive the LD-derivatives of a LLP as a function of its right-hand side. To do this, we use Theorems 6.3.2 and 6.3.3 and apply them to piecewise linear functions defined on closed sets. Then, we extend the results in [60] and apply them to LLPs. Next, we obtain the LD-derivatives of LLPs as function of some elements of their right-hand side. Finally, we use the Phase I LP to obtain an extended system [44]. This extended system provides a way of dealing with LLPs becoming infeasible in the context of optimization and equation solving problems.

Assumption 6.4.1. Let g^i , \mathbf{q}^i and \mathbf{h} for $0 \leq i \leq n_h$ be defined as in Section 2.1 and let Assumption 6.1.1 hold. Assume that $\mathbf{b}_0 \in \text{int}(F)$.

The interior of F is nonempty by Assumption 6.1.1 and Proposition 6.1.2.

Proposition 6.4.4. Let Assumption 6.4.1 hold. Then \mathbf{h} and \mathbf{q}^i are l -smooth at \mathbf{b}_0 for $0 \leq i \leq n_h$. In addition, \mathbf{h} and \mathbf{q}^i are piecewise linear functions on F for all i .

Proof: From [74] we know that piecewise differentiable functions in the sense of Scholtes (see Section 4.1 in [119]) are l -smooth on the interior of their domains. Piecewise linear functions are piecewise differentiable functions. Therefore, the proof shows that for $0 \leq i \leq n_h$, h_i is piecewise linear. All functions g^i are piecewise linear and convex on their respective domains F_i [12]. \mathbf{q}^0 is a linear function on F , therefore, h_0 is piecewise linear on F as it is the composition of a linear function with a piecewise linear function. Now assume that for $i \geq 1$, h_{i-1} and \mathbf{q}_{i-1} are piecewise linear on F . \mathbf{q}^i is a piecewise linear function on F because both h_{i-1} and \mathbf{q}_{i-1} are piecewise linear on F . Then, h_i is piecewise linear on F because it results from the composition of piecewise linear functions. Since h_0 and \mathbf{q}^0 are piecewise linear on

F , it follows by induction that h_i and \mathbf{q}^i are piecewise linear on F for all i . Since $\mathbf{b}_0 \in \text{int}(F)$, both \mathbf{h} and \mathbf{q}^i are l -smooth at \mathbf{b}_0 for $0 \leq i \leq n_h$. \square

Proposition 6.4.5. For $i > 0$ let $F \subset \mathbb{R}^m$ and $F_i \subset \mathbb{R}^{m+i}$ be closed, $g^i : F_i \rightarrow \mathbb{R}$ be piecewise linear and $\mathbf{q}^i : F \rightarrow \text{bnd}(F_i)$ be piecewise affine. Let $h_i = g^i \circ \mathbf{q}^i$. Then for $\mathbf{b}_0 \in \text{int}(F)$, g^i is \mathbf{q}^i -weakly l -smooth at \mathbf{b}_0 and for $\mathbf{M} \in \mathbb{R}^{m \times q}$,

$$h'_i(\mathbf{b}_0; \mathbf{M}) = [g^i]'(\mathbf{q}^i(\mathbf{b}_0); [\mathbf{q}^i]'(\mathbf{b}_0; \mathbf{M})).$$

Proof: h_i is piecewise affine as it results from the composition of a piecewise affine function with a piecewise linear function. Then, h_i and \mathbf{q}^i are both l -smooth at \mathbf{b}_0 because they are PC^1 functions. To apply Theorem 6.3.3, for any $\mathbf{d} \in \mathbb{R}^m$ we need to find $\delta_{\mathbf{d}} > 0$ such that for any $\tau \in (0, \delta_{\mathbf{d}})$, $\mathbf{q}^i(\mathbf{b}_0) + \tau[\mathbf{q}^i]'(\mathbf{b}_0; \mathbf{d}) \in F_i$. By Remark 6.3.3, the rest of the assumptions are satisfied because \mathbf{q}^i are PC^1 functions for all i . From Proposition 6.2.2, since \mathbf{q}^i is piecewise affine there exists $\delta_{\mathbf{d}}^*$ such that for any $\epsilon \in [0, \delta_{\mathbf{d}}^*]$,

$$\mathbf{q}^i(\mathbf{b}_0 + \epsilon \mathbf{d}) = \mathbf{q}^i(\mathbf{b}_0) + \epsilon[\mathbf{q}^i]'(\mathbf{b}_0; \mathbf{d}).$$

Since for all ϵ such that $\mathbf{b}_0 + \epsilon \mathbf{d} \in F$, $\mathbf{q}^i(\mathbf{b}_0 + \epsilon \mathbf{d}) \in F_i$, we can set $\delta_{\mathbf{d}} = \delta_{\mathbf{d}}^*$. Then, Theorem 6.3.3 is applicable and

$$h'_i(\mathbf{b}_0; \mathbf{M}) = [g^i]'(\mathbf{q}^i(\mathbf{b}_0); [\mathbf{q}^i]'(\mathbf{b}_0; \mathbf{M})).$$

In addition, Theorem 6.3.3 establishes that g^i is \mathbf{q}^i -weakly l -smooth at \mathbf{b}_0 . \square

Under Assumption 6.4.1, Theorem 3.3 in [60] gives the LD-derivatives of h_0 for

$\mathbf{M} \in \mathbb{R}^{m \times q}$:

$$\begin{aligned}
[h_0]_{\mathbf{b}_0, \mathbf{M}}^{(j)}(\mathbf{d}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^m} \left[[\mathbf{q}^0]_{\mathbf{b}_0, \mathbf{M}}^{(j)}(\mathbf{d}) \right]^T \boldsymbol{\lambda}, & (6.13) \\
\text{s.t.} \quad & \mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}_0, \\
& -\mathbf{q}^0(\mathbf{b}_0)^T \boldsymbol{\lambda} \leq -h_0(\mathbf{b}_0), \\
& -\left[[\mathbf{q}^0]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \right]^T \boldsymbol{\lambda} \leq -[h_0]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1), \\
& \quad \vdots \\
& -\left[[\mathbf{q}^0]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right]^T \boldsymbol{\lambda} \leq -[h_0]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j).
\end{aligned}$$

Proposition 6.4.6. Let Assumption 6.1.1 hold and let $g : F \rightarrow \mathbb{R} : \mathbf{z} \mapsto g^0(\mathbf{z})$. Let $\mathbf{z}_0 \in F$ and $\mathbf{d} \in \mathbb{R}^m$ be such that there exists $\delta > 0$ such that for all $\epsilon \in [0, \delta)$, $\mathbf{z}_0 + \epsilon \mathbf{d} \in F$. Then,

$$\begin{aligned}
g'(\mathbf{z}_0; \mathbf{d}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^m} \mathbf{d}^T \boldsymbol{\lambda}, & (6.14) \\
\text{s.t.} \quad & \mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}_0, \\
& -\mathbf{z}_0^T \boldsymbol{\lambda} \leq -g(\mathbf{z}_0).
\end{aligned}$$

Proof: Notice that \mathbf{z}_0 is not required to be in the interior of F . g is a convex and piecewise linear function on F [12] of the form $g(\mathbf{z}) = \max_{\boldsymbol{\lambda} \in \Lambda} \mathbf{z}^T \boldsymbol{\lambda}$ where Λ is the finite set that contains all extreme points of the polyhedron $\mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}_0$. Λ is nonempty because \mathbf{A} is full row rank.

Let $\tilde{g} : \mathbb{R}^m \rightarrow \mathbb{R} : \mathbf{z} \mapsto \max_{\boldsymbol{\lambda} \in \Lambda} \mathbf{z}^T \boldsymbol{\lambda}$. Let $\mathbf{z} \in \mathbb{R}^m$. From Proposition 2.2.7 in [27] since \tilde{g} is a convex function that is Lipschitz near \mathbf{z} , $\partial \tilde{g}(\mathbf{z})$ coincides with the subdifferential at \mathbf{z} and for all $\mathbf{d} \in \mathbb{R}^m$, $\tilde{g}'(\mathbf{z}; \mathbf{d}) = \tilde{g}^\circ(\mathbf{z}; \mathbf{d})$. If $\tilde{J}(\mathbf{z}) = \{\boldsymbol{\lambda} \in \Lambda : \tilde{g}(\mathbf{z}) = \mathbf{z}^T \boldsymbol{\lambda}\}$ and $J(\mathbf{z}) = \{\boldsymbol{\lambda} \in \Lambda : g(\mathbf{z}) = \mathbf{z}^T \boldsymbol{\lambda}\}$, it is clear that $\tilde{J}(\mathbf{z}) = J(\mathbf{z})$, $\forall \mathbf{z} \in F$. Since \tilde{g} is the pointwise maximum of convex differentiable functions, $\partial \tilde{g}(\mathbf{z}) = \text{co}(\tilde{J}(\mathbf{z}))$ [15] and for $\mathbf{z} \in F$, $\text{co}(\tilde{J}(\mathbf{z})) = \text{co}(J(\mathbf{z})) = \{\boldsymbol{\lambda} : \mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}_0, \mathbf{z}^T \boldsymbol{\lambda} = g(\mathbf{z})\} = \{\boldsymbol{\lambda} : \mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}_0, -\mathbf{z}^T \boldsymbol{\lambda} \leq -g(\mathbf{z})\}$.

From Proposition 2.1.2 in [27] for $\mathbf{z} \in F$, $\tilde{g}'(\mathbf{z}; \mathbf{d}) = \tilde{g}^\circ(\mathbf{z}; \mathbf{d}) = \max\{\mathbf{d}^\top \boldsymbol{\lambda} : \boldsymbol{\lambda} \in \partial \tilde{g}(\mathbf{z})\} = \max\{\mathbf{d}^\top \boldsymbol{\lambda} : \mathbf{A}^\top \boldsymbol{\lambda} \leq \mathbf{c}_0, -\mathbf{z}^\top \boldsymbol{\lambda} \leq -g(\mathbf{z})\}$. For $\mathbf{z} \in F$ and $\mathbf{d} \in \mathbb{R}^m$ such that there exists $\delta > 0$ such that for all $\epsilon \in [0, \delta)$, $\mathbf{z} + \epsilon \mathbf{d} \in F$, $\tilde{g}'(\mathbf{z}; \mathbf{d}) = g'(\mathbf{z}; \mathbf{d})$. Then, $g'(\mathbf{z}_0; \mathbf{d})$ is given by LP (6.14). \square

Proposition 6.4.7. Let Assumption 6.4.1 hold. Then for $i \in \{0, \dots, n_h\}$ and $j \in \{0, \dots, q\}$ and $\mathbf{M} \in \mathbb{R}^{m \times q}$, the LD-derivatives of \mathbf{h} at \mathbf{b}_0 are given by

$$\begin{aligned} [h_i]_{\mathbf{b}_0, \mathbf{M}}^{(j)}(\mathbf{d}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^{m+i}} \left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(j)}(\mathbf{d}) \right]^\top \boldsymbol{\lambda}, & (6.15) \\ \text{s.t.} \quad & [\mathbf{A}^i]^\top \boldsymbol{\lambda} \leq \mathbf{c}_i, \\ & -\mathbf{q}^i(\mathbf{b}_0)^\top \boldsymbol{\lambda} \leq -h_i(\mathbf{b}_0), \\ & -\left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \right]^\top \boldsymbol{\lambda} \leq -[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1), \\ & \quad \vdots \\ & -\left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right]^\top \boldsymbol{\lambda} \leq -[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j). \end{aligned}$$

Proof: The case for $i = 0$ is established in Theorem 3.3 in [60]. From Proposition 6.4.4, \mathbf{h} and \mathbf{q}^i are piecewise linear and l -smooth at \mathbf{b}_0 for all i . For $i > 0$, by Propositions 6.4.4 and 6.4.5, $h'_i(\mathbf{b}_0; \mathbf{M}) = [g^i]'(\mathbf{q}^i(\mathbf{b}_0); [\mathbf{q}^i]'(\mathbf{b}_0; \mathbf{M}))$. The case for all i and $j = 0$ is established by strong duality of LPs [12] and Proposition 6.4.6; just substitute $F = F_i$, $\mathbf{A} = \mathbf{A}^i$ and $\mathbf{c}_0 = \mathbf{c}_i$, and therefore for all i , $[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{d})$ is given by LP (6.15).

Now assume that for all i and for $j \in \{1, \dots, q\}$, $[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{d})$ is given by LP (6.15). Since for all i , g^i is \mathbf{q}^i -weakly l -smooth at \mathbf{b}_0 , the assumptions of Proposition 6.4.6 are satisfied and $[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(j)}(\mathbf{d})$ is given by LP (6.15). Just substitute,

$$\begin{aligned} \mathbf{A} &= \left[\mathbf{A}^i \quad -\mathbf{q}^i(\mathbf{b}_0) \quad -\left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \right] \cdots - \left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right] \right], \\ \mathbf{c}_0^\top &= \left[\mathbf{c}_i^\top \quad -h_i(\mathbf{b}_0) \quad -[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \quad \cdots \quad -[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right], \\ F &= (G_i)_{\mathbf{b}_0}^{(j)}, \end{aligned}$$

where $(G_i)_{\mathbf{b}_0}^{(j)}$ corresponds to the sets in Definition 6.3.2. Since the case for $j = 0$ is established, the proof follows by induction. \square

Notice that as we calculate the LD-derivatives of LLPs with LP (6.15), we are optimizing over the optimal solution set of (6.12).

Definition 6.4.2. Let Assumption 6.4.1 hold, let $\mathbf{M} \in \mathbb{R}^{m \times q}$ and let $S_0^i(\mathbf{b}_0, \mathbf{M})$ be the solution set of LPs (6.1) and (6.2) and $S_k^i(\mathbf{b}_0, \mathbf{M})$ be the solution set of LP (6.15) for $j = k - 1$, $k \in \{1, \dots, q + 1\}$.

Remark 6.4.1. Let Assumption 6.4.1 hold. Since \mathbf{h} is l -smooth at \mathbf{b}_0 , for any $\mathbf{M} \in \mathbb{R}^{m \times q}$, $\mathbf{d} \in \mathbb{R}^m$, $0 \leq j \leq q$, $-\infty < \mathbf{h}_{\mathbf{b}_0, \mathbf{M}}^{(j)}(\mathbf{d}) < +\infty$. Then for $i \in \{0, \dots, n_h\}$ the LD-derivatives of \mathbf{h} are also given by the primal version of LP (6.15) and for $j \in \{0, \dots, q\}$

$$\begin{aligned}
[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(j)}(\mathbf{d}) = & \\
\min_{\mathbf{v} \in \mathbb{R}^{n_v + j + 1}} & \left[\mathbf{c}_i^T \quad -h_i(\mathbf{b}_0) \quad -[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \quad \dots \quad -[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right] \mathbf{v}, \quad (6.16) \\
\text{s.t.} & \left[\mathbf{A}^i \quad -\mathbf{q}^i(\mathbf{b}_0) \quad -[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \quad \dots \quad -[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right] \mathbf{v} = [\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(j)}(\mathbf{d}), \\
& \mathbf{v} \geq \mathbf{0}.
\end{aligned}$$

Corollary 6.4.1. If Assumption 6.4.1 holds, the LD-derivative of \mathbf{h} at \mathbf{b}_0 in the directions $\mathbf{M} \in \mathbb{R}^{m \times q}$ is given by

$$\mathbf{h}'(\mathbf{b}_0; \mathbf{M}) = \begin{bmatrix} \boldsymbol{\lambda}_0^T \mathbf{m}_1 & \boldsymbol{\lambda}_0^T \mathbf{m}_2 & \dots & \boldsymbol{\lambda}_0^T \mathbf{m}_q \\ \boldsymbol{\lambda}_1^T \left[[\mathbf{q}^1]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \right] & \boldsymbol{\lambda}_1^T \left[[\mathbf{q}^1]_{\mathbf{b}_0, \mathbf{M}}^{(1)}(\mathbf{m}_2) \right] & \dots & \boldsymbol{\lambda}_1^T \left[[\mathbf{q}^1]_{\mathbf{b}_0, \mathbf{M}}^{(q-1)}(\mathbf{m}_q) \right] \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\lambda}_{n_h}^T \left[[\mathbf{q}^{n_h}]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \right] & \boldsymbol{\lambda}_{n_h}^T \left[[\mathbf{q}^{n_h}]_{\mathbf{b}_0, \mathbf{M}}^{(1)}(\mathbf{m}_2) \right] & \dots & \boldsymbol{\lambda}_{n_h}^T \left[[\mathbf{q}^{n_h}]_{\mathbf{b}_0, \mathbf{M}}^{(q-1)}(\mathbf{m}_q) \right] \end{bmatrix}$$

with $\boldsymbol{\lambda}_i \in S_q^i(\mathbf{b}_0, \mathbf{M})$ for $0 \leq i \leq n_h$.

Proof: The definition of LD-derivative is

$$\begin{aligned} h'_i(\mathbf{b}_0; \mathbf{M}) &= \left[[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(q)}(\mathbf{m}_1) [h_i]_{\mathbf{b}_0, \mathbf{M}}^{(q)}(\mathbf{m}_2) \cdots [h_i]_{\mathbf{b}_0, \mathbf{M}}^{(q)}(\mathbf{m}_q) \right], \\ &= \left[[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) [h_i]_{\mathbf{b}_0, \mathbf{M}}^{(1)}(\mathbf{m}_2) \cdots [h_i]_{\mathbf{b}_0, \mathbf{M}}^{(q-1)}(\mathbf{m}_q) \right]. \end{aligned}$$

The second equality follows from Equation (2.4). For any $\boldsymbol{\lambda}_i \in S_q^i(\mathbf{b}_0, \mathbf{M})$, $[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(q-1)}(\mathbf{m}_q) = \boldsymbol{\lambda}_i^T \left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(q-1)}(\mathbf{m}_q) \right]$. Moreover, $\boldsymbol{\lambda}_i \in S_q^i(\mathbf{b}_0, \mathbf{M}) \subset S_{q-1}^i(\mathbf{b}_0, \mathbf{M})$ and therefore $[h_i]_{\mathbf{b}_0, \mathbf{M}}^{(q-2)}(\mathbf{m}_{q-1}) = \boldsymbol{\lambda}_i^T \left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(q-2)}(\mathbf{m}_{q-1}) \right]$. Following this argument,

$$h'_i(\mathbf{b}_0; \mathbf{M}) = \left[\boldsymbol{\lambda}_i^T \left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) \right] \quad \boldsymbol{\lambda}_i^T \left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(1)}(\mathbf{m}_2) \right] \cdots \boldsymbol{\lambda}_i^T \left[[\mathbf{q}^i]_{\mathbf{b}_0, \mathbf{M}}^{(q-1)}(\mathbf{m}_q) \right] \right].$$

□

Example 6.4.1. Let $\mathbf{h} : \mathbb{R}_+^2 \rightarrow \mathbb{R}^2$ where:

$$\begin{aligned} \mathbf{h}(\mathbf{b}) &= \text{lex min}_{\mathbf{v} \in \mathbb{R}^2} \mathbf{C}^T \mathbf{v} \\ &\text{s.t. } v_1 \leq b_1, \\ &\quad v_1 + v_2 \leq b_2, \\ &\quad v_1, v_2 \geq 0, \end{aligned} \tag{6.17}$$

with $\mathbf{C} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$, which is equivalent to first maximizing v_1 and then maximizing v_2 . For LP (6.17), $F = \{\mathbf{b} : \mathbf{b} \geq \mathbf{0}\}$. Consider $\mathbf{b}_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$. Clearly, $\mathbf{b}_0 \in \text{int}(F)$ and $\mathbf{h}(\mathbf{b}_0) = \begin{bmatrix} -1 & 0 \end{bmatrix}^T$. \mathbf{h} is not differentiable at \mathbf{b}_0 . Proposition 6.4.7 provides a way to calculate LD-derivatives of \mathbf{h} . Consider,

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{M}_3 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{M}_4 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}.$$

It can easily be verified that,

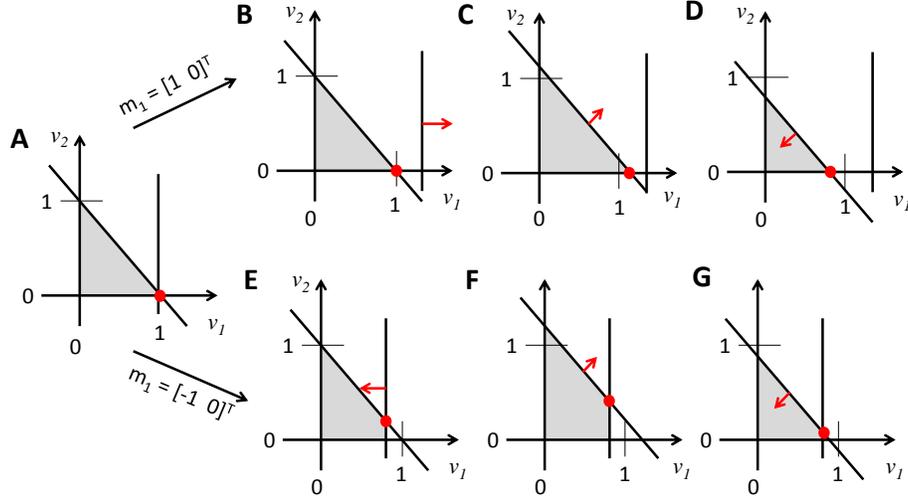


Figure 6-1: Graphical explanation of LD-derivatives for Example 6.4.1. This figure shows graphically how $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$ and \mathbf{M}_4 result in different LD-derivatives at $\mathbf{b}_0 = [1 \ 1]^T$. A) Feasible set (gray) and optimal solution point (red dot) for LP (6.17). If the first column of the matrix of directions is $[1 \ 0]^T$, B is obtained. In B, b_1 is increased and the optimal solution point does not change. Then, the second column of the matrix of directions can be $[0 \ 1]^T$ or $[0 \ -1]^T$ resulting in C and D, respectively. In C, the solution point changes such that h_0 decreases and in D it changes such that h_0 increases. If the first column of the matrix of directions is $[-1 \ 0]^T$, E is obtained. In E, the solution point changes such that h_0 increases and h_1 decreases. Then, the second column of the matrix of directions can be $[0 \ 1]^T$ or $[0 \ -1]^T$ resulting in F and G, respectively. In F, the solution point changes such that h_1 decreases, and in G it changes such that h_1 increases.

$$\mathbf{h}'(\mathbf{b}_0; \mathbf{M}_1) = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, \mathbf{h}'(\mathbf{b}_0; \mathbf{M}_2) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{h}'(\mathbf{b}_0; \mathbf{M}_3) = \begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}, \text{ and } \mathbf{h}'(\mathbf{b}_0; \mathbf{M}_4) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}.$$

From these LD-derivatives, different elements of the lexicographic subdifferential are obtained by solving the system $\mathbf{J}_L \mathbf{h}(\mathbf{b}; \mathbf{M}) \mathbf{M} = \mathbf{h}'(\mathbf{b}; \mathbf{M})$. Then,

$$\mathbf{J}_L \mathbf{h}(\mathbf{b}; \mathbf{M}_1) = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, \mathbf{J}_L \mathbf{h}(\mathbf{b}; \mathbf{M}_2) = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{J}_L \mathbf{h}(\mathbf{b}; \mathbf{M}_3) = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}, \text{ and } \mathbf{J}_L \mathbf{h}(\mathbf{b}; \mathbf{M}_4) = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}.$$

Notice that \mathbf{M}_1 and \mathbf{M}_2 result in the same l -derivative matrix as well as \mathbf{M}_3 and \mathbf{M}_4 . These two matrices form the B-subdifferential of \mathbf{h} at $\mathbf{b}_0 = [1 \ 1]^T$. Proposition 2.6.2 in [27] shows that for a non-scalar function \mathbf{h} evaluated at \mathbf{b}_0 , Clarke's generalized Jacobian is a subset of the Cartesian product of the generalized gradients of each component of \mathbf{h} . In this example, the Cartesian product of the generalized gradients of h_0 and h_1 at $\mathbf{b}_0 = [1 \ 1]^T$ results in the convex hull of $\left\{ \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \right\}$. However, the kinks in the functions h_0 and h_1 are lined up such that the B-subdifferential of \mathbf{h} at $\mathbf{b}_0 = [1 \ 1]^T$ contains only two matrices. The LD-derivatives are guaranteed to find at most these two matrices.

The results of this example can be easily verified by noticing that \mathbf{h} can be expressed as

$$h_0(\mathbf{b}) = -\min\{b_1, b_2\} = \frac{|b_1 - b_2| - b_1 - b_2}{2},$$

$$h_1(\mathbf{b}) = -\max\{0, b_2 - b_1\} = \frac{b_1 - b_2 - |b_1 - b_2|}{2}.$$

Note that so far we considered the optimal values of the LLP to be a function of all right-hand sides of the equality constraints. In practice, we might be interested in the optimal value as a function of a small number of components of the right-hand side. For simplicity, let us suppose that only the first k components of the right-hand side are variable. Hence, for some $k < m$, $k \in \mathbb{N}$, $\mathbf{B} \in \mathbb{R}^{m \times k}$ full column rank, and $\mathbf{b}_0 \in \mathbb{R}^m$ let $\tilde{\mathbf{b}} : \mathbb{R}^k \rightarrow \mathbb{R}^m : \mathbf{u} \mapsto \mathbf{B}\mathbf{u} + \mathbf{b}_0$ and consider the functions for $i \in \{0, \dots, n_h\}$, $\tilde{\mathbf{q}}^i \equiv \mathbf{q}^i \circ \tilde{\mathbf{b}}$ and $\tilde{h}_i \equiv g^i \circ \tilde{\mathbf{q}}^i$. Their domains are given by

$$\tilde{F} \equiv \{\mathbf{u} \in \mathbb{R}^k : -\infty < \tilde{\mathbf{h}}(\mathbf{u}) < +\infty\},$$

which means all components of $\tilde{\mathbf{h}}(\mathbf{u})$ take values in \mathbb{R} . Therefore $\tilde{\mathbf{b}}(\tilde{F}) \subset F$ and for $i \in \{0, \dots, n_h\}$, $\tilde{\mathbf{q}}^i(\tilde{F}) \subset F_i$.

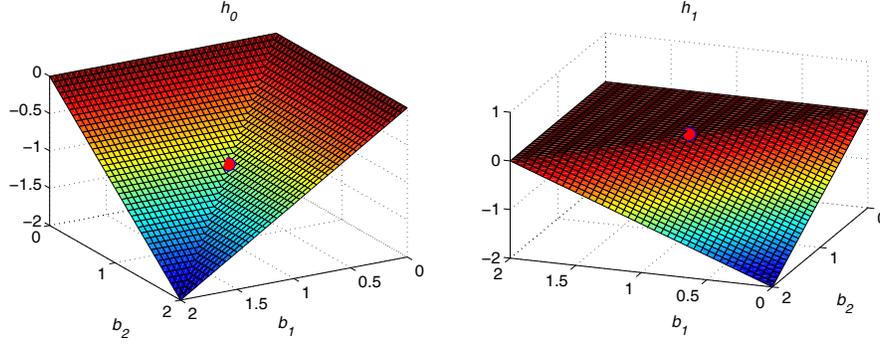


Figure 6-2: Surface plots of \mathbf{h} with respect to \mathbf{b} . The red dots indicate the point $\mathbf{b}_0 = [1 \ 1]^T$. Notice that both components of \mathbf{h} can be divided into two regions of differentiability with two different gradients. In particular, $\nabla h_0(\mathbf{b}) = [-1 \ 0]^T$ or $\nabla h_0(\mathbf{b}) = [0 \ -1]^T$ and $\nabla h_1(\mathbf{b}) = [0 \ 0]^T$ or $\nabla h_1(\mathbf{b})[1 \ -1]^T$. The four matrices of directions $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$ and \mathbf{M}_4 probe possible combinations of these gradients at \mathbf{b}_0 , resulting in two different l -derivative matrices. In fact, these two matrices form the B-subdifferential of \mathbf{h} at \mathbf{b}_0 . In this case, the generalized Jacobian of \mathbf{h} at \mathbf{b}_0 is a strict subset of the Cartesian product of the generalized gradients of each component of \mathbf{h} at \mathbf{b}_0 .

Assumption 6.4.2. Let Assumption 6.1.1 hold. Suppose that $\text{int}(\tilde{F})$ is nonempty, and that $\mathbf{u}_0 \in \text{int}(\tilde{F})$.

Computing LD-derivatives of $\tilde{\mathbf{h}}$ can be challenging when $\mathbf{B}\mathbf{u} + \mathbf{b}_0$ is in the boundary of F . For example, consider LP (6.17) and let $\mathbf{B} = [0 \ 1]^T$ and $\mathbf{b}_0 = \mathbf{0}$. For such $\tilde{\mathbf{b}}$, $\tilde{F} = \{u : u \geq 0\}$, and $\forall u \in \tilde{F}$, $\tilde{\mathbf{b}}(u) \in \text{bnd}(F)$. Therefore for this situation, the LD-derivatives of $\tilde{\mathbf{h}}$ can't be computed using the chain rule in Equation 2.5. However, the extensions of LD-derivatives presented in Theorems 6.3.2 and 6.3.3 can help us compute the LD-derivatives for this case. We shall now show how to compute directional and LD-derivatives of $\tilde{\mathbf{h}}$.

Proposition 6.4.8. Let Assumption 6.4.2 hold. Then for $i \in \{0, \dots, n_h\}$, $\tilde{\mathbf{h}}$ and $\tilde{\mathbf{q}}^i$ are l -smooth at \mathbf{u}_0 and piecewise affine on \tilde{F} .

Proof: From Proposition 6.4.4, \mathbf{h} and \mathbf{q}^i are piecewise linear on F for all i . Since $\tilde{\mathbf{b}}$ is piecewise affine on \mathbb{R}^k , $\tilde{\mathbf{h}}$ and $\tilde{\mathbf{q}}^i$ are piecewise affine on \tilde{F} for all i . Therefore, they are piecewise differentiable functions in the sense of Scholtes [119] and l -smooth [74] at \mathbf{u}_0 . \square

The following Remark is analogous to Proposition 6.4.7. The case for $\tilde{\mathbf{b}}(\mathbf{u}_0) \in$

$\text{int}(F)$ follows directly from the chain rule. If $\tilde{\mathbf{b}}(\mathbf{u}_0) \in \text{bnd}(F)$, then it follows from Propositions 6.4.8, 6.4.5 and 6.4.6 and the proof is very similar to the one in Proposition 6.4.7.

Remark 6.4.2. Let Assumption 6.4.2 hold at $\mathbf{u} = \mathbf{u}_0$. Then for any $\mathbf{d} \in \mathbb{R}^k$, $q \in \mathbb{N}$, $\mathbf{M} \in \mathbb{R}^{k \times q}$, and $i \in \{0, \dots, n_h\}$, and $j \in \{0, \dots, q\}$,

$$\begin{aligned}
[\tilde{h}_i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^{m+i}} \left[[\tilde{\mathbf{q}}^i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d}) \right]^T \boldsymbol{\lambda}, & (6.18) \\
\text{s.t.} \quad & [\mathbf{A}^i]^T \boldsymbol{\lambda} \leq \mathbf{c}_i, \\
& -\tilde{\mathbf{q}}^i(\mathbf{u})^T \boldsymbol{\lambda} \leq -\tilde{h}_i(\mathbf{u}), \\
& - \left[[\tilde{\mathbf{q}}^i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{m}_1) \right]^T \boldsymbol{\lambda} \leq -[\tilde{h}_i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{m}_1), \\
& \quad \vdots \\
& - \left[[\tilde{\mathbf{q}}^i]_{\mathbf{u}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right]^T \boldsymbol{\lambda} \leq -[\tilde{h}_i]_{\mathbf{u}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j).
\end{aligned}$$

6.4.2 Phase I LP as an extended system

LLPs present complications when they become infeasible. When this happens, DFBA simulations, optimization algorithms or nonsmooth equation solving methods fail. To deal with this problem, the Phase I LP of the Simplex algorithm can be used to extend the domain of \mathbf{h} because it provides an alternative LLP that is always feasible [44]. In particular, when the LLP presented in LPs (6.1) and (6.2) is feasible, the extended system given by the Phase I LP and the original system coincide. Otherwise, the extended system is still defined and provides a penalty function [44]. Setting the penalty function equal to zero can be added as a constraint to the optimization problem or as an equation in nonsmooth equation solving algorithms.

Definition 6.4.3. Consider the LP (6.1). A Phase I LP of (6.1) is given by [12]:

$$\begin{aligned} \min_{\mathbf{v} \in \mathbb{R}^{n_v}, \mathbf{s}_+, \mathbf{s}_- \in \mathbb{R}^m} \quad & \sum_{i=1}^m s_{+i} + s_{-i}, \\ \text{s.t.} \quad & \mathbf{A}\mathbf{v} + \mathbf{s}_+ - \mathbf{s}_- = \mathbf{z}, \\ & \mathbf{v} \geq \mathbf{0}, \mathbf{s}_+ \geq \mathbf{0}, \mathbf{s}_- \geq \mathbf{0}. \end{aligned} \tag{6.19}$$

It is a well-known fact that the Phase I LP of (6.1) is a LP that is feasible for any $\mathbf{z} \in \mathbb{R}^m$ and its objective function value is equal to zero if and only if (6.1) is feasible and positive otherwise.

Proposition 6.4.9. Let Assumption 6.1.1 hold. Now let $\mathbf{h}^E = [h_{-1}^E \quad h_0^E \quad \dots \quad h_{n_h}^E]^\top : \mathbb{R}^m \rightarrow \mathbb{R}^{n_h+2}$:

$$\begin{aligned} h_{-1}^E(\mathbf{z}) = \min_{\mathbf{v} \in \mathbb{R}^{n_v}, \mathbf{s}_+, \mathbf{s}_- \in \mathbb{R}^m} \quad & \sum_{i=1}^m s_{+i} + s_{-i}, \\ \text{s.t.} \quad & \mathbf{A}\mathbf{v} + \mathbf{s}_+ - \mathbf{s}_- = \mathbf{z}, \\ & \mathbf{v} \geq \mathbf{0}, \mathbf{s}_+ \geq \mathbf{0}, \mathbf{s}_- \geq \mathbf{0}. \end{aligned}$$

and for $0 \leq i \leq n_h$

$$\begin{aligned} h_i^E(\mathbf{z}) = \min_{\mathbf{v} \in \mathbb{R}^{n_v}, \mathbf{s}_+, \mathbf{s}_- \in \mathbb{R}^m} \quad & \mathbf{c}_i^\top \mathbf{v}, \\ \text{s.t.} \quad & \mathbf{A}\mathbf{v} + \mathbf{s}_+ - \mathbf{s}_- = \mathbf{z}, \\ & \sum_{i=1}^m s_{+i} + s_{-i} = h_{-1}^E(\mathbf{z}), \\ & \begin{bmatrix} \mathbf{c}_0^\top \\ \vdots \\ \mathbf{c}_{i-1}^\top \end{bmatrix} \mathbf{v} = \begin{bmatrix} h_0^E(\mathbf{z}) \\ \vdots \\ h_{i-1}^E(\mathbf{z}) \end{bmatrix}, \\ & \mathbf{v} \geq \mathbf{0}, \mathbf{s}_+ \geq \mathbf{0}, \mathbf{s}_- \geq \mathbf{0}. \end{aligned} \tag{6.20}$$

Then \mathbf{h}^E is l -smooth on \mathbb{R}^m . If $h_{-1}^E(\mathbf{z}) = 0$, then LPs (6.1) and (6.2) are feasible and $h_i^E(\mathbf{z}) = h_i(\mathbf{z})$ for $0 \leq i \leq n_h$.

Proof: Under Assumption 6.1.1 F is nonempty. If $h_{-1}^E(\mathbf{z}) = \sum_{i=1}^m s_{+i} + s_{-i} = 0$, then $\mathbf{z} \in F$ because there exists $\mathbf{v} \geq \mathbf{0}$ such that $\mathbf{A}\mathbf{v} = \mathbf{z}$. Then LP (6.1) is feasible and by Proposition (6.1.3) LP (6.2) is also feasible for all i . If $h_{-1}^E(\mathbf{z}) = 0$, then $\mathbf{s}_+, \mathbf{s}_- = \mathbf{0}$, and the variables $\mathbf{s}_+, \mathbf{s}_-$ and the constraint $\sum_{i=1}^m s_{+i} + s_{-i} = h_{-1}^E(\mathbf{z})$ can be removed from the LPs (6.20) for $0 \leq i \leq n_h$. Then $h_i^E(\mathbf{z}) = h_i(\mathbf{z})$ for $0 \leq i \leq n_h$. By Proposition 6.1.2, F is nonempty. Therefore, for $\mathbf{z} \in F$, $\mathbf{h}^E(\mathbf{z}) \in \mathbb{R}^{n_h+2}$. This implies that the dual LPs of (6.19) and (6.20) are always feasible.

Now let us show that (6.19) and (6.20) satisfy Assumption 6.4.1. Let $n_h^p = n_h + 1$, $\mathbf{A}^p = \begin{bmatrix} \mathbf{A} & \mathbf{I}_m & -\mathbf{I}_m \end{bmatrix}$. Let $(\mathbf{c}_0^p)^\top = \begin{bmatrix} \mathbf{0}^\top & \mathbf{1}^\top & \mathbf{1}^\top \end{bmatrix}$ and for $i \in \{0, \dots, n_h\}$, let $(\mathbf{c}_{i+1}^p)^\top = \begin{bmatrix} \mathbf{c}_i^\top & \mathbf{0}^\top & \mathbf{0}^\top \end{bmatrix}$. Then LPs (6.19) and (6.20) can be expressed in the format of LPs (6.1) and (6.2) by letting $n_h = n_h^p$, $\mathbf{A} = \mathbf{A}^p$ and for all i , $\mathbf{c}_i = \mathbf{c}_i^p$. Since the dual LPs of (6.19) and (6.20) are always feasible, \mathbf{A}^p and \mathbf{c}_i^p , $i = 0, \dots, n_h^p$ are such that $\mathbf{h}^E(\mathbf{z}) > -\infty$ for all $\mathbf{z} \in \mathbb{R}^m$. Since \mathbf{A} is full row rank, \mathbf{A}_p is full row rank. Then LPs (6.19) and (6.20) satisfy Assumption 6.4.1. Then by Proposition 6.4.4, \mathbf{h}^E is l -smooth for any $\mathbf{z} \in \mathbb{R}^m$. \square

Let $\tilde{\mathbf{h}}^E = \mathbf{h}^E \circ \tilde{\mathbf{b}}$. Since, $\tilde{h}_i^E(\mathbf{u}) = \tilde{h}_i(\mathbf{u})$ for $0 \leq i \leq n_h$ for $\mathbf{u} \in \tilde{F}$, the LD-derivatives of \tilde{h}_i^E and \tilde{h}_i for $0 \leq i \leq n_h$ coincide on $\text{int}(\tilde{F})$. Then, $\tilde{\mathbf{h}}^E$ can be used to calculate the LD-derivatives of $\tilde{\mathbf{h}}$.

6.5 Implementation of LD-derivatives in nonsmooth equation solving algorithms

Next we present three examples implementing the LD-derivatives of lexicographic linear programs to solve two nonsmooth equation solving problems and a nonsmooth optimization problem. All running times for the next two examples are for a 3.20 GHz Intel®; Xeon®; CPU in MATLAB 7.12 (R2011a), Windows 7 64-bit operating system using 4 processors for computations in parallel. The LP solvers used were CPLEX [28] and Gurobi [50].

Example 6.5.1. This example is taken from [22, 23]. In these papers, fermentation of synthesis gas to ethanol and acetate takes place in a bubble column bioreactor with syngas fermenting bacterium *Clostridium Ijungdahlii*. This is a new technology that is being considered for production of biofuels from natural gas. This bubble column bioreactor can be modeled by the following partial differential equation (PDE) system:

1. Mass balance of biomass of *C. Ijungdahlii*:

$$\begin{aligned} \frac{\partial X}{\partial t}(z, t) &= \mu(z, t)X(z, t) - \frac{u_L}{\epsilon_L} \frac{\partial X}{\partial z}(z, t) + D_A \frac{\partial^2 X}{\partial z^2}(z, t), \\ u_L X(0, t) - \epsilon_L D_A \frac{\partial X}{\partial z}(0, t) &= 0, \quad \frac{\partial X}{\partial z}(L, t) = 0, \quad X(z, 0) = X_0, \end{aligned}$$

where X is the concentration of biomass, t is time, μ is the growth rate, u_L is the liquid velocity, z is the spatial position, D_A is the diffusivity, ϵ_L is the liquid volume fraction in the reactor, and X_0 is the initial biomass concentration.

2. Mole balances of liquid-phase CO, H₂, CO₂, ethanol, and acetate:

$$\begin{aligned} \frac{\partial G_L}{\partial t}(z, t) &= \\ v_G(z, t)X(z, t) + \frac{k_{m,G}}{\epsilon_L}(G^*(z, t) - G_L(z, t)) - \frac{u_L}{\epsilon_L} \frac{\partial G_L}{\partial z}(z, t) + D_A \frac{\partial^2 G_L}{\partial z^2}(z, t), \\ u_L G_L(0, t) - \epsilon_L D_A \frac{\partial G_L}{\partial z}(0, t) &= u_L G_{gF} H_G, \quad \frac{\partial G_L}{\partial z}(L, t) = 0, \quad G_L(z, 0) = G_{L0}, \end{aligned}$$

where G can be CO, H₂, CO₂, ethanol, and acetate concentrations, v_G refers to the exchange flux rate for species G , $k_{m,G}$ the liquid mass transfer coefficient for species G , G_L the liquid concentration of species G , G^* the liquid concentration in equilibrium with the gas concentration of species G , G_{gF} the gas concentration in the feed of species G , G_{L0} the initial concentration of species G in the liquid, and H_G is Henry's constant for species G .

3. Mole balances of gas-phase CO, H₂, and CO₂:

$$\frac{\partial G_g}{\partial t}(z, t) = -\frac{k_{m,G}}{\epsilon_g}(G^*(z, t) - G_L(z, t)) - \frac{u_g}{\epsilon_g} \frac{\partial G_g}{\partial z}(z, t),$$

$$G_g(0, t) = G_{gF}, G_g(z, 0) = G_{g0},$$

where G_g is the concentration of species G in the gas, ϵ_g is the gas volume fraction, u_g is the gas velocity, and G_{g0} is the initial gas concentration of species G .

4. Column pressure profile:

$$P(z) = P_L + \rho_L g(L - z),$$

where P is the pressure as a function of position in the column, P_L is the pressure at the top of the column, ρ_L is the density of the liquid, L is the size of the column, and g is gravitational acceleration.

To obtain the growth rate μ and the exchange flux rates v_G , this problem is transformed into a DFBA problem. Following the strategy in [59], a hierarchy of objectives is established in Table 6.1.

Table 6.1: Hierarchy of objectives for bubble column bioreactor

1	Minimize slacks in Phase I feasibility LP
2	Maximize growth
3	Maximize CO uptake
4	Maximize H ₂ uptake
5	Minimize CO ₂ production
6	Minimize acetate production
7	Minimize ethanol production

The uptake kinetics for CO, H₂ and CO₂ are described by the following equation:

$$v_G = \frac{v_{max,G}G}{K_{m,G} + G} \frac{1}{1 + \frac{E_L + A_L}{K_I}}.$$

These uptake kinetics provide some upper bounds to the exchange flux rates in the

FBA problem.

The goal is to compute the steady state of this system. One way is to run the dynamic simulation for a long time. Alternatively, a nonsmooth system of equations can be solved by setting all time derivatives to zero. To solve this system of equations, sensitivity information is needed and this is where the LD-derivatives of LLPs come into play. The resulting nonsmooth system of equations were solved for the following parameter values:

$$\begin{aligned}
u_g &= 75 \text{ m/h}, L = 25\text{m}, u_L = 0.25 \text{ m/h}, D_A = 0.25 \text{ m}^2/\text{h}, T = 310.15\text{K}, \\
P_L &= 1 \text{ atm}, \rho_L = 1000 \text{ kg/m}^3, P_{CO} = 0.6 * P_0, P_{H_2} = 0.4 * P_0, \\
G_{gF} &= \frac{P_G}{R * T}, H_{CO} = 8.0 \times 10^{-4} \frac{\text{mol}}{(L * \text{atm})}, H_{H_2} = 6.6 \times 10^{-4} \frac{\text{mol}}{(L * \text{atm})}, \\
H_{CO_2} &= 2.5 \times 10^{-2} \frac{\text{mol}}{(L * \text{atm})}, k_{m,CO_2} = k_{m,CO} = 80/\text{h}, k_{m,H_2} = 2.5k_{m,CO}, \\
k_{m,E} &= k_{m,A} = 0, \epsilon_g = u_g \frac{0.53}{3600(0.15 + u_g/3600)}, \epsilon_L = 1 - \epsilon_g, \\
v_{max,CO} &= 35 \frac{\text{mmol}}{\text{g} * \text{h}}, v_{max,CO} = v_{max,CO_2} = 35 \frac{\text{mmol}}{\text{g} * \text{h}}, v_{max,H_2} = 70 \frac{\text{mmol}}{\text{g} * \text{h}}, \\
K_{m,CO} &= K_{m,CO_2} = 0.02\text{mmol/L}, K_{m,H_2} = 0.02\text{mmol/L}, K_I = 10\text{mmol/L},
\end{aligned}$$

where P_0 is the pressure at the bottom of the column, P_G is the partial pressure of species G , T is the temperature in Kelvin, and R is the universal gas constant. Using finite volumes to discretize the spatial dimension of the bubble column, 100 nodes were considered. The states vector for each finite volume is provided in the following order: [biomass, CO_L , CO_g , H_{2g} , H_{2L} , CO_{2L} , CO_{2g} , A, E], where A stands for acetate and E for ethanol. This results in a system comprising 901 equations (the last equation corresponds to the penalty), 100 LLPs each one with 682 equality constraints, 1715 variables, and 7 objective functions. Three different strategies were used to obtain sensitivity information for this system of equations:

1. LD-derivatives (LD) in the directions **I**,
2. Directional derivatives (DD) in the coordinate directions; note that these are not guaranteed to be B-subdifferential elements,

3. Finite differences (FD).

Notice that whereas for the LD method, LP (6.18) is being solved for all i and all j , in the DD method LP (6.18) is solved only for $j = 0$ (only directional derivatives are computed). Therefore, the LPs being solved to find the LD-derivatives have smaller feasible sets than the ones being solved to find the directional derivative in each coordinate direction. At nonsmooth points, this can result in the DD method not returning an element of the B-subdifferential (the LD method always returns elements of the B-subdifferential).

The nonsmooth Newton method [103] was used to solve this system. The LLP associated to *Clostridium Ijungdahlii* satisfies Assumption 6.4.1. Two different starting points were considered and the method converged to two different solution points: washout and the non-trivial solution. All finite volumes used the same starting point; therefore, only the starting vector for a single finite volume is reported. Tables 6.4 and 6.5 present the number of iterations, the 2-norm of the residual vector and the total time for each method.

Table 6.2: Number of iterations and 2-norm reported for each method with a start point of $[0.1, 1.6421, 80.6372, 0.9032, 53.7581, 0, 0, 0, 0]$ for each finite volume.

Iteration	LD	DD	FD
1	71.9045	71.9045	71.9045
2	3.6064×10^{-10}	3.6819×10^{-10}	0.0025
3			3.6032×10^{-5}
4			6.3648×10^{-9}
Time(s)	242.6	374.72	319.9
Result	Washout	Washout	Washout

Finding the steady state using the dynamic simulation takes 1462.0 seconds from the first start point and 6629.8 seconds from the second start point. Both dynamic simulations converge to the non-trivial steady state that supports growth and fermentation of the syngas (as opposed to the washout steady state). It can be noticed that finding the analytical derivatives results in less iterations than finite differences. Also, given that the LPs solved by LD-derivatives have smaller feasible sets than the ones solved by finding the directional derivatives in each coordinate direction, the

Table 6.3: Number of iterations and 2-norm reported for each method with a start point of [20, 0.5, 50, 0, 50, 0, 0, 10, 50] for each finite volume.

Iteration	LD, 2-norm	DD, 2-norm	FD, 2-norm
1	1.448×10^5	1.448×10^5	1.448×10^5
2	1.074×10^3	1.074×10^3	1.074×10^3
3	567.5	567.5	567.5
4	251.1	251.1	251.1
5	12.47	12.47	12.47
6	1.305	1.305	1.306
7	0.0114	0.0114	0.0115
8	8.416×10^{-7}	8.416×10^{-7}	1.098×10^{-5}
9			1.283×10^{-7}
Time(s)	2873.9	4115.7	783.0
Result	Non-trivial solution	Non-trivial solution	Non-trivial solution

computation of the steady state using LD-derivatives is faster. A sure indication that a nondifferentiable point has been encountered is provided when the LD and DD iterations differ.

Example 6.5.2. The behavior of the firms in this example are adapted from Example 16-3 in [63]. Consider a firm that produces two products. Both products require a mix of three chemicals. The supply of Chemicals 1 and 2 cannot be exceeded, and that of Chemical 3 is flexible but it is desired to keep it below 6 units because of safety concerns. Chemical 1 is significantly more difficult to handle than Chemical 2. A profit of at least \$600 is needed or the firm will be closed. The lexicographic objectives of the firm are:

1. to minimize any use of Chemical 3 above 6 units (h_0^1),
2. to maximize profit (h_1^1),
3. to minimize the total use of Chemicals 1 and 2 (h_2^1),
4. to minimize the use of Chemical 1 (h_3^1).

The profit and chemical requirements for each product are given in Table 6.4.

There is also a second firm that produces two other products with the first two chemicals. This company requires a minimum profit of \$1300 and its lexicographic objectives are:

Table 6.4: Data for Firm 1 in Example 6.5.2

	Profit per unit	Units Chemical 1	Units Chemical 2	Units Chemical 3
Product 1	80	4	4	1
Product 2	100	5	2	0
Supply		b_1	b_2	6^*

* The supply of Chemical 3 is flexible but it is desired to keep its consumption below 6 units.

1. to minimize any Chemical usage exceeding 120 units (h_0^2),
2. to maximize profit (h_1^2),
3. to minimize the total use of Chemicals 1 and 2 (h_2^2),
4. to minimize the use of Chemical 1 (h_3^2).

The data for Firm 2 are presented in Table 6.5.

Table 6.5: Data for Firm 2 in Example 6.5.2

	Profit per unit	Units Chemical 1	Units Chemical 2
Product 1	75	6	2
Product 2	110	3	6
Supply		b_3	b_4

Chemicals 1 and 2 are sold by a single supplier. This supplier produces 180 units of Chemical 1 and 120 units of Chemical 2. This supplier can decide how much of each chemical to sell to each supplier. The supplier has negotiated different price schemes with the two firms which are summarized in Table 6.6. The supplier wants both firms to stay in business to reduce the risk of having a single customer. All remaining Chemicals not sold to Firms 1 or 2 are sold at very low price (50 cents for Chemical 1 and 40 cents for Chemical 2). The goal is for the supplier to decide how much to sell of each chemical to each firm such that it maximizes its own profit.

Let x_1 and x_2 be the number of units produced of Products 1 and 2 by Firm 1 and let v_1 and v_2 be the number of units produced of Products 1 and 2 by Firm 2. All other variables are slack variables. Both LPs describing the behavior of Firms 1 and 2 are converted to standard form and the Phase I extension is implemented. The

Table 6.6: Prices of Chemicals

Firm 1	Chemical 1	Chemical 2
First 30 units	2.50/unit	2/unit
Units 30 to 60	2.25/unit	1.9/unit
Units exceeding 60	2/unit	1.8/unit
Firm 2	Chemical 1	Chemical 2
First 40 units	2.50/unit	2/unit
Units 40 to 80	2.10/unit	1.8/unit
Units exceeding 80	1.80/unit	1.6/unit

optimization problem to be solved can be defined as:

$$\max_{\mathbf{b}} \alpha_1(\mathbf{b}) + \alpha_2(\mathbf{b}) + \alpha_3(\mathbf{b}) + \alpha_4(\mathbf{b}) + 0.5\alpha_5(\mathbf{b}) + 0.4\alpha_6(\mathbf{b}), \quad (6.21)$$

$$\text{s.t. } b_1 + b_3 \leq 180,$$

$$b_2 + b_4 \leq 120,$$

$$h_{-1}^1(\mathbf{b}) + h_{-1}^2(\mathbf{b}) = 0,$$

$$\alpha_1(\mathbf{b}) = \begin{cases} 2.5h_3^1(\mathbf{b}) & \text{if } h_3^1(\mathbf{b}) \leq 30, \\ 75 + 2.25(h_3^1(\mathbf{b}) - 30) & \text{if } 30 < h_3^1(\mathbf{b}) \leq 60, \\ 142.5 + 2(h_3^1(\mathbf{b}) - 60) & \text{otherwise,} \end{cases}$$

$$\alpha_2(\mathbf{b}) = \begin{cases} 2.5h_3^2(\mathbf{b}) & \text{if } h_3^2(\mathbf{b}) \leq 40, \\ 100 + 2.1(h_3^2(\mathbf{b}) - 40) & \text{if } 40 < h_3^2(\mathbf{b}) \leq 80, \\ 184 + 1.8(h_3^2(\mathbf{b}) - 80) & \text{otherwise,} \end{cases}$$

$$\alpha_3(\mathbf{b}) = \begin{cases} 2(h_2^1(\mathbf{b}) - h_3^1(\mathbf{b})) & \text{if } h_2^1(\mathbf{b}) - h_3^1(\mathbf{b}) \leq 30, \\ 60 + 1.9(h_2^1(\mathbf{b}) - h_3^1(\mathbf{b}) - 30) & \text{if } 30 < h_2^1(\mathbf{b}) - h_3^1(\mathbf{b}) \leq 60, \\ 117 + 1.8(h_2^1(\mathbf{b}) - h_3^1(\mathbf{b}) - 60) & \text{otherwise,} \end{cases}$$

$$\alpha_4(\mathbf{b}) = \begin{cases} 2(h_2^2(\mathbf{b}) - h_3^2(\mathbf{b})) & \text{if } h_2^2(\mathbf{b}) - h_3^2(\mathbf{b}) \leq 40, \\ 80 + 1.8(h_2^2(\mathbf{b}) - h_3^2(\mathbf{b}) - 40) & \text{if } 40 < h_2^2(\mathbf{b}) - h_3^2(\mathbf{b}) \leq 80, \\ 152 + 1.6(h_2^2(\mathbf{b}) - h_3^2(\mathbf{b}) - 80) & \text{otherwise,} \end{cases}$$

$$\alpha_5(\mathbf{b}) = 180 - b_1 - b_3, \quad \alpha_6(\mathbf{b}) = 120 - b_2 - b_4,$$

where $\mathbf{h}^1(\mathbf{b})$ and $\mathbf{h}^2(\mathbf{b})$ are obtained from the solution of the LLPs that describe the behavior of Firms 1 and 2. The behavior of Firm 1 can be modeled as:

$$\begin{aligned} \mathbf{h}^1(\mathbf{b}) &= \text{lex min}_{\mathbf{x} \in \mathbb{R}^{10}} \mathbf{C}^T \mathbf{x} \\ \text{s.t. } & 4x_1 + 5x_2 + x_3 - x_4 = b_1, \\ & 4x_1 + 2x_2 + x_5 - x_6 = b_2, \\ & 80x_1 + 100x_2 - x_7 + x_8 = 600, \\ & x_1 + x_9 - x_{10} = 6, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where

$$\mathbf{C}^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -80 & -100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and the behavior of Firm 2 can be modeled as:

$$\begin{aligned} \mathbf{h}^2(\mathbf{b}) &= \text{lex min}_{\mathbf{v} \in \mathbb{R}^{10}} \mathbf{C}^T \mathbf{v} \\ \text{s.t. } & 6v_1 + 3v_2 + v_3 - v_4 = b_3, \\ & 2v_1 + 6v_2 + v_5 - v_6 = b_4, \\ & 75v_1 + 110v_2 - v_7 + v_8 = 1000, \\ & 8v_1 + 9v_2 + v_9 - v_{10} = 120, \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned}$$

where

$$\mathbf{C}^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -75 & -110 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This optimization problem was solved using Solvopt [80] in MATLAB. The feasibility and optimality tolerance of the LP solver was 1×10^{-7} . The termination criteria for Solvopt was a feasibility tolerance of 1×10^{-8} , a relative error on the function value of 1×10^{-6} and a relative error on the iterates of 1×10^{-4} based on the infinity-norm. This problem was solved without providing any Jacobian information and providing an element of the B-subdifferential of the objective function and constraints using LD-derivatives. Running times are presented in Table 6.7. The use of LD-derivatives reduced running time and increased reliability.

Table 6.7: Running times for the optimization problem (6.21).

$\mathbf{b}_f = (128.6, 51.4, 51.4, 68.6)$, Fval = 638.9

\mathbf{b}_i	No Jacobian information		Jacobian from LD-derivatives	
	Iterations	t[s]	Iterations	t[s]
(0,0,0,0)	80	49.0	55	23.7
(-10,50,100,-1000)	66	39.7	59	25.5
(80,60,100,60)	78	53.7	51	21.5

Example 6.5.3. Consider a continuous stirred-tank reactor (CSTR) with a volume of 1L and a dilution rate of 0.1 L/h. This reactor has a controlled concentration of oxygen of 7.7 mg/L, and glucose and xylose are fed at a concentration of 5 and 1 g/L, respectively. *E. Coli* is inoculated in the reactor. The goal is to solve for the steady-state concentrations of biomass, glucose, xylose, and ethanol. The metabolic network of *E. Coli* is available from [107]. This metabolic network consists of 1075 reactions and 761 metabolites. This problem can be solved by simulating the DFBA problem until its steady state is approximately reached using DFBAlab [44]. For instance, if we start the simulation at concentrations of 0.001, 0.1, 0.1 and 0 g/L

of biomass, glucose, xylose, and ethanol, respectively, DFBAlab takes 1.4 seconds to simulate 5000 hours and find the steady-state of 2.35, 3.4×10^{-4} , 8.8×10^{-4} , 0 g/L of biomass, glucose, xylose, and ethanol, respectively. Instead, the steady-state can be found by solving a nonsmooth system of equations. The system of equations results from finding the concentrations of biomass, glucose, xylose, and ethanol that make the right-hand side of the ODE describing the CSTR equal to zero.

The system of equations solved is:

$$\begin{aligned} f_1(\mathbf{x}) &= D(x_b^f - x_b)/V + \mu(\mathbf{x})x_b = 0, \\ f_j(\mathbf{x}) &= D(x_i^f - x_i)/V + v_i(\mathbf{x})x_b = 0, \text{ for } i = g, x, e \text{ and } j = 2, 3, 4, \\ p(\mathbf{x}) &= 0, \end{aligned}$$

where b , g , x and e refer to biomass, glucose, xylose, and ethanol, respectively, x_i^f is the concentration rate in the feed, x_i is the concentration in the CSTR, $\mu(\mathbf{x})$ is the growth rate, $\mathbf{v}(\mathbf{x}) = [v_g(\mathbf{x}) \ v_x(\mathbf{x}) \ v_e(\mathbf{x})]^T$ are the exchange fluxes, $p(\mathbf{x})$ is the penalty state obtained from the Phase I LP, and $\mathbf{x} = [x_b \ x_g \ x_x \ x_e]^T$. $\mu(\mathbf{x})$, $\mathbf{v}(\mathbf{x})$, and $p(\mathbf{x})$ are obtained from the solution of a LLP. The LP in standard form contains 753 equality constraints and 2100 variables. The lexicographic order of objectives is the following:

1. minimize penalty,
2. maximize growth,
3. maximize ethanol production,
4. minimize glucose consumption,
5. minimize xylose consumption.

To solve this problem, we used five methods:

1. `fsolve` in MATLAB without providing derivative information (FSND),

2. `fsolve` in MATLAB providing an element of the B-subdifferential using LD-derivatives, (FSWD),
3. the classical Newton method approximating the Jacobian with finite differences using $\epsilon = 1 \times 10^{-6}$ (NFD),
4. the quasi-Newton method described in [78] providing an element of the B-subdifferential using LD-derivatives (QSNM),
5. the ∞ -norm version of the LP-Newton method described in [33] providing an element of the B-subdifferential using LD-derivatives (LPNM).

The LP feasibility and optimality tolerances for the FBA model were of 1×10^{-9} . For the FSND and FSWD methods, the parameter associated to function value tolerance ‘TolFun’ was set equal to 1×10^{-8} . The rest of the methods were assumed to have converged if $\|\mathbf{f}(\mathbf{x})\| < 1 \times 10^{-8}$. The LP-Newton method allows to constrain the solution to a convex polyhedral set. The constraints imposed made all concentrations nonnegative. In some instances, biomass was constrained to be greater or equal than 1 g/L (marked with an asterisk).

We solved the problem ignoring the last equation $p(\mathbf{x}) = 0$ to obtain a square system and we verified that this equation was satisfied after a solution was found. For all solution points \mathbf{x}^* found by the different algorithms, $p(\mathbf{x}^*) = 0$. The running times and steady-state results are presented in Table 6.8. For the first start vector, all methods find the same steady-state solution. The use of LD-derivatives increases the speed of `fsolve` and the quasi-Newton method is the fastest closely followed by the NFD method. The next two start vectors lead to different steady-state solutions. When the start vector is (1,1,1,1), the washout solution (no biomass in the CSTR) attracts all methods, but the constraints imposed on the LP-Newton method help find the non-trivial steady-state solution.

To eliminate the possibility of washout, a feed of 0.1 g/L of E.Coli was imposed on the system. DFBAlab takes 1.4 seconds to simulate 5000 hours and find the steady-state of 2.43, 3.3×10^{-4} , 8.5×10^{-4} , 0 g/L of biomass, glucose, xylose, and ethanol,

respectively. The results of this modified simulation are presented in the lower part of Table 6.8. Surprisingly, all first four methods are unable to find the non-trivial steady-state solution and instead find a steady state that is not feasible (negative biomass concentrations). Once again, the constraints on the LP-Newton method help locate the non-trivial steady-state solution. The LP-Newton method fails in one case in which it cycles. The cycling occurs because at the start point of (1,1,1,1), the method enters the region of attraction of the washout solution. Therefore, the LP-Newton method stays at the boundary of the convex polyhedral set and is unable to satisfy the termination criteria. This cycling is avoided by adding the constraint of biomass being ≥ 1 g/L which allows the iterates to escape the region of attraction of the washout solution.

Table 6.8: Located solutions and running times for a CSTR non-smooth equation solving problem.

No Feed *E. Coli*.

Solutions: A:($2.35, 3.4 \times 10^{-4}, 8.8 \times 10^{-4}, 0$), B:(0,5,1,0)

Start	FSND	FSWD	NFD	QSNM	LPNM
(0,0,0,0)	B, 2.5s	B, 1.9s	B, 0.5s, 1	B, 0.7s, 1	B, 5.0s, 12
(2,0,0,0)	A, 3.5s	A, 2.4s	B, 1.5s, 3	B, 1.0s, 2	A, 2.6s, 7
(2,1,1,0)	B, 2.6s	B, 2.1s	B, 1.7s, 3	B, 1.6s, 3	A, 4.2s, 11
(1,1,1,1)	B, 2.6s	B, 2.1s	B, 1.7s, 3	B, 1.6s, 3	B, 5.8s, 13 A, 4.7s, 12*

Feed *E. Coli* = 0.1 g/L

Solutions: A:($2.43, 3.3 \times 10^{-4}, 8.5 \times 10^{-4}, 0$), B:(-0.015,5.28,1,0)

Start	FSND	FSWD	NFD	QSNM	LPNM
(2,1,1,0)	B, 2.6s	B, 2.0s	B, 1.7s, 3	B, 1.6s, 3	A, 4.1s, 11
(1,1,1,1)	B, 2.6s	B, 2.1s	B, 1.7s, 3	B, 1.6s, 3	Fails A, 4.3s, 12*

Note: The asterisk means that a constraint making biomass greater or equal than 1 g/L was imposed on the LP Newton method. The last number on the last three columns indicates the number of iterations.

Notice also that the NFD and QSNM methods take almost the same time in most cases to find a solution. This is because both methods are very similar when the LLP is smooth at all iterates; finite differences will approximate the Jacobian and LD-derivatives will equal the Jacobian (within the LP solution tolerance). Let $\mathbf{x}_{(k)}^1$ and $\mathbf{x}_{(k)}^2$ be the k^{th} iterates of the NFD and QSNM methods, respectively. In fact, for

all cases except when the start point is $(2,0,0,0)$, both methods have the same number of iterates and for any k and any starting point, $\|\mathbf{x}_{(k)}^1 - \mathbf{x}_{(k)}^2\| < 5 \times 10^{-7}$. When the method starts at $(0,0,0,0)$, the QSNM converges to the exact solution whereas the NFD approximates the solution numerically. When, the method starts at $(2,0,0,0)$, both methods present a different number of iterates (see Table 6.9). This suggests that the numerical approximation of the Jacobian obtained from LD-derivatives is more accurate than the one obtained through finite differences.

Table 6.9: Sequence of iterates for the NFD (1) and QSNM (2) when starting at $(2,0,0,0)$.

k	$\mathbf{x}_{(k)}^1$	$\mathbf{x}_{(k)}^2$	$\ \mathbf{x}_{(k)}^1 - \mathbf{x}_{(k)}^2\ $
0	$(2,0,0,0)$	$(2,0,0,0)$	0
1	$(5.76 \times 10^{-11}, 3.57 \times 10^{-4}, 9.15 \times 10^{-4}, 0)$	$(0, 3.57 \times 10^{-4}, 9.15 \times 10^{-4}, 0)$	1.43×10^{-7}
2	$(-1.57 \times 10^{-5}, 5.00, 1.00, 0)$	$(0, 5, 1, 0)$	3.7×10^{-5}
3	$(-6.60 \times 10^{-15}, 5.00, 1.00, 0)$		

The LPNM and QSNM exhibit local Q-quadratic convergence in the vicinity of a solution when elements of the B-subdifferential are used [78, 33]. Tables 6.10 and 6.11 show that Q-quadratic convergence is indeed occurring in both methods.

Table 6.10: Sequence of iterates for the QSNM with no *E. Coli* feed.

$k \setminus \mathbf{x}_{(0)}$	$(0,0,0,0)$	$(2,0,0,0)$	$\ \mathbf{x}_{(k)} - \mathbf{x}^*\ $ $(2,1,1,0)$	$(1 \ 1 \ 1 \ 1)$
0	5.099	5.477	4.4721	4.2426
1	0	5.0985	0.4612	0.5980
2		0	2.2169×10^{-6}	4.6659×10^{-5}
3			1.1102×10^{-16}	1.8062×10^{-14}

In both examples, the use of LD-derivatives increased speed. In the nonsmooth equation solving problem, LD-derivatives provide useful derivative information for the LP-Newton method which allows us to find solutions restricted to a convex polyhedral set with local Q-quadratic convergence rate. Also, the LP-Newton method is able to handle rectangular systems. Table 6.12 shows the running times using FSND, FSWD, and LPNM and Table 6.13 shows that the iterates of the LPNM exhibit Q-quadratic convergence when the equation $p(\mathbf{x}) = 0$ is incorporated in the system of equations being solved.

Table 6.11: Sequence of iterates for the LPNM with no *E. Coli* feed.

$k \setminus \mathbf{x}_{(0)}$	(0,0,0,0)	(2,0,0,0)	$\ \mathbf{x}_{(k)} - \mathbf{x}^*\ $ (2,1,1,0)	(1,1,1,1)	(1 1 1 1)*
0	5.0990	0.3525	1.4567	4.2426	2.1970
1	4.2850	0.6859	1.2637	4.6887	2.3221
2	4.0503	0.5587	1.8350	5.0773	3.0927
3	3.2420	0.2293	1.8884	4.4970	2.6835
4	2.4823	6.82×10^{-2}	1.5475	3.6350	2.1579
5	1.7527	4.02×10^{-3}	0.9069	2.8292	1.7185
6	1.1026	2.63×10^{-5}	0.6816	2.0789	0.9999
7	0.5691	7.60×10^{-10}	0.2656	1.3886	0.7236
8	0.2020		9.06×10^{-2}	0.7960	0.2879
9	3.30×10^{-2}		6.64×10^{-3}	0.3463	0.1059
10	1.02×10^{-3}		7.46×10^{-5}	8.69×10^{-2}	8.70×10^{-3}
11	1.01×10^{-6}		5.94×10^{-9}	6.74×10^{-3}	1.31×10^{-4}
12	3.47×10^{-12}			4.37×10^{-5}	1.80×10^{-8}
13				1.84×10^{-9}	

Note: The asterisk means that the constraint of biomass greater or equal than 1 g/L was imposed.

6.6 Conclusions

LLPs are useful to model bioprocesses using FBA and DFBA or business decisions involving goal-programming. In both cases, nonsmooth optimization and equation-solving problems embedding LLPs can be formulated. To solve these problems, generalized derivative information for LLPs is desirable. this chapter obtains elements of the B-subdifferential of the values function of a LLP as a function of its right-hand side by solving a number of related LPs. The two examples presented illustrate how LD-derivatives can be used to increase the solution speed and reliability of nonsmooth equation and optimization problems embedding LLPs. This work opens the possibility of optimizing DFBA systems as shown in Chapters 7 and 8.

Table 6.12: Located solutions and running times for a CSTR non-smooth equation solving problem.

No Feed *E. Coli*.

Solutions: A:($2.35, 3.4 \times 10^{-4}, 8.8 \times 10^{-4}, 0$), B:(0,5,1,0)

Start	FSND	FSWD	LPNM
(0,0,0,0)	B, 4.8s	B, 4.3s	B, 5.6s, 13
(2,0,0,0)	A, 3.0s	A, 2.2s	A, 2.7s, 7
(2,1,1,0)	B, 3.7s	B, 3.2s	A, 4.2s, 11
(1,1,1,1)	B, 3.7s	B, 3.2s	B, 5.6s, 13 A, 4.4s, 12*

Note: The asterisk means that a constraint making biomass greater or equal than 1 g/L was imposed on the LP Newton method. The last number on the last three columns indicates the number of iterations.

Table 6.13: Sequence of iterates for LPNM with no *E. Coli* feed.

k \ $\mathbf{x}_{(0)}$	(0,0,0,0)	(2,0,0,0)	$\ \mathbf{x}_{(k)} - \mathbf{x}^*\ $ (2,1,1,0)	(1,1,1,1)	(1 1 1 1)*
0	5.0990	0.3525	1.4567	4.2426	2.1970
1	5.0948	0.6859	1.2637	4.6887	2.3221
2	4.7745	0.5587	1.8350	5.0773	3.0927
3	4.2695	0.2293	1.8884	4.4970	2.6835
4	3.5006	6.82×10^{-2}	1.5475	3.6350	2.1579
5	2.7201	4.02×10^{-3}	0.9069	2.8292	1.7185
6	1.9713	2.63×10^{-5}	0.6816	2.0789	0.9999
7	1.2933	4.17×10^{-10}	0.2656	1.3886	0.7236
8	0.7188		9.06×10^{-2}	0.7960	0.2879
9	0.2948		6.64×10^{-3}	0.3463	0.1059
10	6.54×10^{-2}		7.46×10^{-5}	8.69×10^{-2}	8.70×10^{-3}
11	3.89×10^{-3}		3.50×10^{-9}	6.74×10^{-3}	1.31×10^{-4}
12	1.46×10^{-5}			4.37×10^{-5}	1.06×10^{-8}
13	2.06×10^{-10}			1.84×10^{-9}	

Note: The asterisk means that the constraint biomass ≥ 1 g/L was imposed.

Chapter 7

Sensitivities of Dynamic Flux Balance Analysis Models

7.1 Introduction

Consider a parameter-dependent system of ordinary differential equations (ODEs) of the form

$$\begin{aligned}\dot{\mathbf{x}}(t, \mathbf{p}) &= \mathbf{f}(t, \mathbf{x}(t, \mathbf{p}), \mathbf{h}(\mathbf{x}(t, \mathbf{p}))), \quad \forall t \in (t_0, t_f], \\ \mathbf{x}(t_0, \mathbf{p}) &= \mathbf{p},\end{aligned}\tag{7.1}$$

where $\mathbf{h}(\mathbf{z}) \in \mathbb{R}^{n_h+1}$ is the vector of optimal values of the parametric lexicographic linear program (LLP):

$$\begin{aligned}h_0(\mathbf{z}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_0^T \mathbf{v}, \\ \text{s.t. } \mathbf{A}\mathbf{v} &= \mathbf{b}(\mathbf{z}), \\ \mathbf{v} &\geq \mathbf{0},\end{aligned}\tag{7.2}$$

and for $1 \leq i \leq n_h$

$$\begin{aligned}
h_i(\mathbf{z}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_i^T \mathbf{v}, \\
\text{s.t. } & \begin{bmatrix} \mathbf{A} \\ \mathbf{c}_0^T \\ \vdots \\ \mathbf{c}_{i-1}^T \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{b}(\mathbf{z}) \\ h_0(\mathbf{z}) \\ \vdots \\ h_{i-1}(\mathbf{z}) \end{bmatrix}, \\
& \mathbf{v} \geq \mathbf{0}.
\end{aligned} \tag{7.3}$$

For an open set $D_x \subset \mathbb{R}^{n_x}$, $\mathbf{b} : D_x \rightarrow \mathbb{R}^m$ and $\mathbf{f} : (t_0, t_f] \times D_x \times \mathbb{R}^{n_h+1} \rightarrow \mathbb{R}^{n_x}$ are *abs-factorable* functions [71]. A more compact way of writing LPs (7.2) and (7.3) is:

$$\begin{aligned}
\mathbf{h}(\mathbf{z}) &= \text{lex} \min_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{C}^T \mathbf{v}, \\
\text{s.t. } & \mathbf{A} \mathbf{v} = \mathbf{b}(\mathbf{z}), \\
& \mathbf{v} \geq \mathbf{0},
\end{aligned} \tag{7.4}$$

where $\mathbf{C} = [\mathbf{c}_0 \ \dots \ \mathbf{c}_{n_h}]$. To provide modeling flexibility, we will assume that the function $\mathbf{b} = \tilde{\mathbf{b}} \circ \hat{\mathbf{b}}$, where $\tilde{\mathbf{b}}$ is of the form $\tilde{\mathbf{b}} : \mathbb{R}^k \rightarrow \mathbb{R}^m : \mathbf{z} \mapsto \mathbf{B}\mathbf{z} + \mathbf{b}_0$. We denote Equation (7.1) with \mathbf{h} given by Equations (7.2) and (7.3) as a *dynamic system with a lexicographic linear program embedded*. A broad class of dynamic optimization problems for this type of system can be formulated as:

$$\begin{aligned}
\min_{\mathbf{p}} J(\mathbf{p}) &\equiv \varphi(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} l(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt \\
\text{s.t. } \mathbf{g}(\mathbf{p}) &\equiv \mathbf{r}(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} \mathbf{s}(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt \leq \mathbf{0}, \\
\mathbf{p} &\in P \subset \mathbb{R}^{n_x}.
\end{aligned} \tag{7.5}$$

This includes optimal control problems and optimization of bioreactor models using dynamic flux balance analysis (DFBA) [87, 51, 59]. The solution of this class of optimization problems presents two main difficulties. First, J and \mathbf{g} in (7.5) are usu-

ally not differentiable everywhere. Therefore, numerical methods require generalized derivative information for J and \mathbf{g} with respect to \mathbf{p} . This difficulty is addressed in the work of Höffner, Khan and Barton [60, 72]. In their work, lexicographic differentiation [96] provides a way of obtaining generalized derivative information for J and \mathbf{g} with respect to \mathbf{p} for *dynamic systems with a linear program embedded* using nonsmooth forward sensitivity analysis. Second, for a given $\mathbf{p} \in P$ the solution of (7.1) might not exist for all $t \in [t_0, t_f]$ and therefore, J and \mathbf{g} cannot be evaluated at these points. The extension of the feasible set for the ODE (7.1) using an LLP proposed by Gomez *et al.* [44] combined with a penalty method scheme [83] enables solving (7.5), once reliable generalized derivative information can be obtained.

Generalized derivatives of LLPs have already been derived in [47]. LLPs are particularly useful in the context of *dynamic flux balance analysis* (DFBA). DFBA systems are simulated as *dynamic systems with LLPs embedded* [59, 56]. DFBA takes into account genome-scale information to perform dynamic simulations of microbial communities [137, 87, 51, 59, 58, 45]. The complications of dynamic systems with LLPs embedded has hindered its widespread utilization, but with the simulators in [59] and [44], reliable and efficient simulation of DFBA systems is now possible. The optimization of DFBA systems will enable the optimal design of industrial bioprocesses employing microbial communities.

This chapter extends the work in [60] by obtaining generalized derivative information for ODEs with LLPs embedded. The work in [60] considers LPs in standard form parameterized by their right-hand side whereas this chapter considers LLPs instead. The difference is important as LLPs allow the implementation of a penalty function and ensure uniqueness of the right-hand side of ODE (7.1). Sensitivities for ODE systems with LLPs embedded do not follow trivially from the theory in [60], but require the results in [47] and [72] to be derived. The work in [72] shows that sensitivity information of ODE systems with LLPs embedded can be obtained as the unique solutions of related ODE systems by using the concept of *lexicographic directional derivatives* (LD-derivatives) [74]. However, these related ODE systems do not necessarily satisfy Carathéodory's conditions [35]. The work in [71] presents an algorithm

that uses an event-detection scheme to deal with this difficulty. To apply this work to ODE systems with LLPs embedded, the embedded LLPs need to be expressed as a composition of absolute value and C^1 functions, making this approach intractable. Instead, we use basis information from linear program (LP) solvers to transform the ODE system with LLPs embedded into a sequence of differential-algebraic equation (DAE) systems in time, with an approach similar to the one presented in [56].

7.2 Sensitivities of ODE systems with LLPs embedded

7.2.1 LD-derivatives of ODE systems with LLPs embedded

Let $\mathbf{x}_t(\mathbf{p}) = \mathbf{x}(t, \mathbf{p})$. To solve (7.5), it is extremely helpful to compute the sensitivities of \mathbf{x}_t which is a nonsmooth function. Here, we use LD-derivatives to represent the sensitivity of \mathbf{x}_t . Let $\tilde{\mathbf{f}}_t(\mathbf{z}) = \mathbf{f}(t, \mathbf{z}, \mathbf{h}(\mathbf{b}(\mathbf{z})))$ and $q \in \mathbb{N}$. Since \mathbf{h} is abs-factorable and l -smooth [47], $\tilde{\mathbf{f}}_t$ is abs-factorable and l -smooth [74]. From Theorem 4.2 in [72] for $\mathbf{M} \in \mathbb{R}^{n_x \times q}$, the LD-derivatives $[\mathbf{x}_t]'(\mathbf{p}; \mathbf{M})$ are the unique solution of the following ODE system:

$$\frac{d\mathbf{S}}{dt}(t) = [\tilde{\mathbf{f}}_t]'(\mathbf{x}(t, \mathbf{p}); \mathbf{S}(t)), \quad \mathbf{S}(t_0) = \mathbf{M}. \quad (7.6)$$

ODE (7.6) does not necessarily satisfy Carathéodory's conditions because $[\mathbf{f}_t]'(\mathbf{x}(t, \mathbf{p}); \cdot)$ can be a discontinuous function (see Example 4.1 in [72]). However, the columns of this matrix ODE can be decoupled into Carathéodory ODEs [72]. Based on this observation, an event detection scheme can be used to divide ODE (7.6) into a sequence of Carathéodory ODEs [73, 71]. To apply the numerical method in [71], an explicit abs-factorable representation of $[\tilde{\mathbf{f}}_t]'$ is needed. To obtain $[\tilde{\mathbf{f}}_t]'$, $[\mathbf{h}]'$ is needed which is obtained from solving more LLPs [47]. Since expressing LLPs explicitly as abs-factorable functions is intractable, we will develop an approach that does not require $[\tilde{\mathbf{f}}_t]'$ to be expressed explicitly as an abs-factorable function. In the next sections we

develop the numerical method necessary to obtain $[\mathbf{x}_t]'(\mathbf{p}; \mathbf{M})$.

7.2.2 LD-derivatives of LLPs using the approach in Chapter 6

First of all, let us define formally the LLP and introduce the LLPs that need to be solved to obtain the LD-derivatives of a LLP. Consider a LLP parameterized by its right-hand side. For each $i \in \{0, 1, \dots, n_h\}$, let $g^i : \mathbb{R}^{m+i} \rightarrow \overline{\mathbb{R}}$, where for $\mathbf{z} \in \mathbb{R}^m$:

$$\begin{aligned} g^0(\mathbf{z}) &= \inf_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_0^T \mathbf{v}, \\ \text{s.t. } \mathbf{A} \mathbf{v} &= \mathbf{z}, \\ \mathbf{v} &\geq \mathbf{0}, \end{aligned} \tag{7.7}$$

and for $1 \leq i \leq n_h$ and $\mathbf{z} \in \mathbb{R}^{m+i}$:

$$\begin{aligned} g^i(\mathbf{z}) &= \inf_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_i^T \mathbf{v}, \\ \text{s.t. } \begin{bmatrix} \mathbf{A} \\ \mathbf{c}_0^T \\ \vdots \\ \mathbf{c}_{i-1}^T \end{bmatrix} \mathbf{v} &= \mathbf{z}, \\ \mathbf{v} &\geq \mathbf{0}. \end{aligned} \tag{7.8}$$

Assumption 7.2.1. Let \mathbf{A} be of full row rank. For all $i \in \{0, \dots, n_h\}$, let \mathbf{A} and \mathbf{c}_j , $j = 0, \dots, i$, be such that $g^i(\mathbf{z}) > -\infty$ for all $\mathbf{z} \in \mathbb{R}^{m+i}$. In addition for all $i > 0$, let $\begin{bmatrix} \mathbf{A}^T & \mathbf{c}_0 & \dots & \mathbf{c}_{i-1} \end{bmatrix}^T$ be full row rank.

Let $F = \{\mathbf{z} \in D_x : -\infty < g^0(\mathbf{b}(\mathbf{z})) < +\infty\}$, $\tilde{F} = \{\mathbf{z} \in \mathbb{R}^k : -\infty < g^0(\tilde{\mathbf{b}}(\mathbf{z})) < +\infty\}$, $\hat{F} = \mathbf{b}(F) \subset \mathbb{R}^m$, $\mathbf{A}^0 = \mathbf{A}$ and for $i \geq 1$ let $\mathbf{A}^i = \begin{bmatrix} \mathbf{A}^T & \mathbf{c}_0 & \dots & \mathbf{c}_{i-1} \end{bmatrix}^T$. It is clear that $\hat{\mathbf{b}}(F) \subset \tilde{F}$. Under Assumption 7.2.1, \mathbf{h} in (7.2) and (7.3) is such that $\mathbf{h} : F \rightarrow \mathbb{R}^{n_h+1}$. In this chapter, only LLPs satisfying Assumption 7.2.1 will be considered.

Definition 7.2.1. Let Assumption 7.2.1 hold. Let $\mathbf{q}^i : F \rightarrow \mathbb{R}^{m+i}$ where $\mathbf{q}^0 : \mathbf{z} \mapsto \mathbf{b}(\mathbf{z})$ and for $i = 1, \dots, n_h$, $\mathbf{q}^i : \mathbf{z} \mapsto \left[\mathbf{b}(\mathbf{z})^\top \ h_0(\mathbf{z}) \ \dots \ h_{i-1}(\mathbf{z}) \right]^\top$. Let $\mathbf{h} : F \rightarrow \mathbb{R}^{n_h+1}$, which contains the objective function values in LPs (7.2) and (7.3) when $\mathbf{z} \in F$, and is such that $h_i = g^i \circ \mathbf{q}^i$.

If \mathbf{A} is full row rank, but \mathbf{A}^{n_h} is not full row rank, the linearly dependent cost vectors can be eliminated from the respective technology matrices \mathbf{A}^i and right-hand sides \mathbf{q}^i with $i \in \{1, \dots, n_h\}$.

Remark 6.4.2 combined with the chain rule for LD-derivatives indicate that for $\mathbf{M} \in \mathbb{R}^{n_x \times q}$, $\mathbf{d} \in \mathbb{R}^{n_x}$, $j \in \{0, \dots, q\}$, and $i \in \{0, \dots, n_h\}$, the LD-derivatives of \mathbf{h} at $\mathbf{u} \in \text{int}(F)$ when $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\tilde{F})$ can be computed by solving the following LP:

$$\begin{aligned}
[h_i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^{m+i}} \left[[\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d}) \right]^\top \boldsymbol{\lambda}, & (7.9) \\
\text{s.t.} \quad & [\mathbf{A}^i]^\top \boldsymbol{\lambda} \leq \mathbf{c}_i, \\
& -\mathbf{q}^i(\mathbf{u})^\top \boldsymbol{\lambda} \leq -h_i(\mathbf{u}), \\
& -\left[[\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{m}_1) \right]^\top \boldsymbol{\lambda} \leq -[h_i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{m}_1), \\
& \quad \quad \quad \vdots \\
& -\left[[\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right]^\top \boldsymbol{\lambda} \leq -[h_i]_{\mathbf{u}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j),
\end{aligned}$$

Remark 6.4.1 takes the dual of Equation (7.9) which results in

$$\begin{aligned}
[h_i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v+j+1}} \left[\mathbf{c}_i^\top \ -h_i(\mathbf{u}) \ -[h_i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{m}_1) \ \dots \ -[h_i]_{\mathbf{u}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right] \mathbf{v}, & (7.10) \\
\text{s.t.} \quad & \left[\mathbf{A}^i \ -\mathbf{q}^i(\mathbf{u}) \ -[\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{m}_1) \ \dots \ -[\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right] \mathbf{v} = [\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d}), \\
& \mathbf{v} \geq \mathbf{0}.
\end{aligned}$$

Remark 7.2.1. For $\mathbf{M} \in \mathbb{R}^{n_x \times q}$, $\mathbf{d} \in \mathbb{R}^{n_x}$, and $j \in \{0, \dots, q\}$ the LD-derivatives of

\mathbf{h} at $\mathbf{u} \in \text{int}(F)$ when $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\widetilde{F})$ are given by the following LLP:

$$\begin{aligned} [\mathbf{h}]_{\mathbf{u},\mathbf{M}}^{(j)}(\mathbf{d}) = \text{lex} \min_{\mathbf{v} \in \mathbb{R}^{n_v+j+1}} & [\mathbf{C}_{\mathbf{u},\mathbf{M}}^{(j)}]^\text{T} \mathbf{v}, \\ \text{s.t. } & \mathbf{A}_{\mathbf{u},\mathbf{M}}^{(j)} \mathbf{v} = [\mathbf{q}^0]_{\mathbf{u},\mathbf{M}}^{(j)}(\mathbf{d}), \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{7.11}$$

where $[\mathbf{C}_{\mathbf{u},\mathbf{M}}^{(j)}]^\text{T} \equiv \left[[\mathbf{C}_{\mathbf{u},\mathbf{M}}^{(j-1)}]^\text{T} \quad -[\mathbf{h}]_{\mathbf{u},\mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right]$ and $[\mathbf{C}_{\mathbf{u},\mathbf{M}}^{(0)}]^\text{T} \equiv \left[\mathbf{C}^\text{T} \quad -\mathbf{h}(\mathbf{u}) \right]$, and $\mathbf{A}_{\mathbf{u},\mathbf{M}}^{(j)} = \left[\mathbf{A}_{\mathbf{u},\mathbf{M}}^{(j-1)} \quad -[\mathbf{q}^0]_{\mathbf{u},\mathbf{M}}^{(j-1)}(\mathbf{m}_j) \right]$ and $\mathbf{A}_{\mathbf{u},\mathbf{M}}^{(0)} = \left[\mathbf{A} \quad -\mathbf{q}^0(\mathbf{u}) \right]$.

Notice that if the Phase I LP is used as the level zero of the LLP as suggested in [47], $F = D_x$, $\widetilde{F} = \mathbb{R}^k$ and then the requirement that $\mathbf{u} \in \text{int}(F)$ and that $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\widetilde{F})$ can easily be satisfied.

This approach of computing the LD-derivatives of \mathbf{h} , though theoretically sound, has posed challenging numerical difficulties. Although this approach works most of the times, it does fail a few instances. It is not rare for state-of-the-art LP solvers such as CPLEX [28] or Gurobi [50] to fail, finding LLP (7.11) to be infeasible, unbounded, or complaining about numerical problems, sometimes even contradicting each other. In particular, the cost vectors and the resulting technology matrices in LLP (7.11) contain objective function values from other LPs. This objective function values are guaranteed to be within some tolerance $\epsilon > 0$. Small errors in elements that become part of the technology matrix can cause the solution of LLP (7.11) to fail in unpredictable ways. This behavior has motivated us to explore alternative ways to compute the LD-derivatives of LLPs.

7.2.3 Alternative methods to compute LD-derivatives of LLPs

Theorem 2 in [56] provides a method to compute efficiently and reliably $\mathbf{h}(\mathbf{z})$ for any given $\mathbf{z} \in F$. In particular, Algorithm 2 in [56] provides a way to obtain an optimal basis B for the LLP. This basis can be used to compute \mathbf{h} for different values of $\mathbf{z} \in F$ using Equation (12) in [56] as long as the basis remains feasible, which is verified by the following condition: $\mathbf{A}_B^{-1} \mathbf{b}(\mathbf{z}) \geq \mathbf{0}$. Therefore, $\mathbf{h}(\mathbf{z}) = \mathbf{C}_B^\text{T} \mathbf{A}_B^{-1} \mathbf{b}(\mathbf{z})$.

Given that an LLP parameterized by its right-hand side is an abs-factorable function, \mathbf{h} is abs-factorable, and therefore piecewise differentiable (\mathcal{PC}^1). The l -derivatives of \mathcal{PC}^1 functions are guaranteed to be in the B-subdifferential (limiting Jacobian) of \mathbf{h} [74]. By Proposition 4.1.3 in [119], the B-subdifferential of \mathcal{PC}^1 functions contain the Jacobian matrices of all essentially active functions. In the context of LLPs, it may be difficult to determine whether a basis matrix results in an essentially active function. Nevertheless, LD-derivatives may still be computed if the respective piece and the directions matrix \mathbf{M} satisfy certain properties. The following Propositions and Theorem provide a way of calculating LD-derivatives of piecewise affine functions in a more efficient way.

Proposition 7.2.1. Let $D_x \in \mathbb{R}^{n_x}$ be open and convex, $\mathbf{f} : D_x \rightarrow \mathbb{R}^n$ be \mathcal{PC}^1 with selection functions $\mathbf{f}_{(i)} : D_x \rightarrow \mathbb{R}^n$ with $i \in \{1, \dots, n_i\}$ and let $\mathbf{M} \in \mathbb{R}^{n_x \times q}$. For $i \in \{1, \dots, n_i\}$, let $D_x^i = \{\mathbf{x} \in D_x : \mathbf{f}(\mathbf{x}) = \mathbf{f}_{(i)}(\mathbf{x})\}$. Consider $\mathbf{z} \in D_x$ and let $j \in \{1, \dots, n_i\}$ be such that $\mathbf{f}(\mathbf{z}) = \mathbf{f}_{(j)}(\mathbf{z})$. Assume that there exists $\delta \in (0, 1)$ such that for any $\epsilon \in (0, \delta)$, $\mathbf{f}(\mathbf{z} + \epsilon \mathbf{m}_1) = \mathbf{f}_{(j)}(\mathbf{z} + \epsilon \mathbf{m}_1)$. Then, $\mathbf{f}_{\mathbf{z}, \mathbf{M}}^{(0)}(\mathbf{m}_1) = [\mathbf{f}_{(j)}]_{\mathbf{z}, \mathbf{M}}^{(0)}(\mathbf{m}_1) = \mathbf{J}\mathbf{f}_{(j)}(\mathbf{z})\mathbf{m}_1$.

Proof: From the definition of LD-derivatives,

$$\mathbf{f}_{\mathbf{z}, \mathbf{M}}^{(0)}(\mathbf{m}_1) = \lim_{\tau \rightarrow 0^+} \frac{\mathbf{f}(\mathbf{z} + \tau \mathbf{m}_1) - \mathbf{f}(\mathbf{z})}{\tau}. \quad (7.12)$$

Consider the sequence $\{\tau_n\}$ such that $\tau_1 = \delta/2$ and for $n > 1$, $\tau_n = \tau_{n-1}/2$. This sequence can be used to compute the limit in Equation (7.12). Then,

$$\mathbf{f}_{\mathbf{z}, \mathbf{M}}^{(0)}(\mathbf{m}_1) = \lim_{n \rightarrow \infty} \frac{\mathbf{f}(\mathbf{z} + \tau_n \mathbf{m}_1) - \mathbf{f}(\mathbf{z})}{\tau_n} = \lim_{n \rightarrow \infty} \frac{\mathbf{f}_{(j)}(\mathbf{z} + \tau_n \mathbf{m}_1) - \mathbf{f}_{(j)}(\mathbf{z})}{\tau_n} = \mathbf{J}\mathbf{f}_{(j)}(\mathbf{z})\mathbf{m}_1.$$

□

Proposition 7.2.2. Let $D_x \in \mathbb{R}^{n_x}$ be open and convex, $\mathbf{f} : D_x \rightarrow \mathbb{R}^n$ be \mathcal{PC}^1 with selection functions $\mathbf{f}_{(i)} : D_x \rightarrow \mathbb{R}^n$ with $i \in \{1, \dots, n_i\}$ and let $\mathbf{M} \in \mathbb{R}^{n_x \times q}$. For $i \in \{1, \dots, n_i\}$, let $D_x^i = \{\mathbf{x} \in D_x : \mathbf{f}(\mathbf{x}) = \mathbf{f}_{(i)}(\mathbf{x})\}$. Consider $\mathbf{z} \in D_x$ and let $j \in \{1, \dots, n_i\}$ be such that $\mathbf{f}(\mathbf{z}) = \mathbf{f}_{(j)}(\mathbf{z})$. Let $k \in \{1, \dots, q-1\}$ and assume there

exists $\delta \in (0, 1)$ such that for any $\epsilon \in (0, \delta)$, $\mathbf{f}_{\mathbf{z}, \mathbf{M}}^{(k-1)}(\mathbf{m}_k + \epsilon \mathbf{m}_{k+1}) = [\mathbf{f}_{(j)}]_{\mathbf{z}, \mathbf{M}}^{(k-1)}(\mathbf{m}_k + \epsilon \mathbf{m}_{k+1}) = \mathbf{J}\mathbf{f}_{(j)}(\mathbf{z})[\mathbf{m}_k + \epsilon \mathbf{m}_{k+1}]$. Then, $\mathbf{f}_{\mathbf{z}, \mathbf{M}}^{(k)}(\mathbf{m}_{k+1}) = [\mathbf{f}_{(j)}]_{\mathbf{z}, \mathbf{M}}^{(k)}(\mathbf{m}_{k+1}) = \mathbf{J}\mathbf{f}_{(j)}(\mathbf{z})\mathbf{m}_{k+1}$.

Proof: The proof is very similar to the proof of Proposition 7.2.1. \square

Let $\tilde{\mathbf{h}} : \tilde{F} \rightarrow \mathbb{R}^{n_h+1}$, and for $i \in \{0, \dots, n_h\}$, $\tilde{\mathbf{q}}^i : \tilde{F} \rightarrow \mathbb{R}^{m+i}$ where $\tilde{\mathbf{q}}^0(\mathbf{z}) = \tilde{\mathbf{b}}(\mathbf{z})$ and $\tilde{\mathbf{q}}^k(\mathbf{z}) = [\tilde{\mathbf{q}}^{k-1}(\mathbf{z}) \ \tilde{h}_{k-1}(\mathbf{z})]$ for $k \in \{1, \dots, n_h\}$. Then for all i , $\tilde{h}_i = g^i \circ \tilde{\mathbf{q}}^i$. It is clear that $\tilde{\mathbf{h}}$ is a piecewise affine function. Remember that $\tilde{\mathbf{b}} : \mathbf{z} \mapsto \mathbf{B}\mathbf{z} + \mathbf{b}_0$.

Theorem 7.2.3. Consider $\tilde{\mathbf{h}}$ and let $\mathbf{z}_0 \in \text{int}(\tilde{F})$. Assume that the basis B is optimal in the sense of Algorithm 2 in [56] for $\tilde{\mathbf{h}}$ at \mathbf{z}_0 . Let $\mathbf{M} \in \mathbb{R}^{k \times q}$. Let the function

$$f\text{sign} : \mathbb{R}^l \rightarrow \mathbb{R} : \mathbf{x} \mapsto \begin{cases} 0 & \text{if } \mathbf{x} = \mathbf{0}, \\ 1 & \text{if the first nonzero component of the vector is positive} \\ -1 & \text{otherwise.} \end{cases}$$

Let $\mathbf{K} = \mathbf{A}_B^{-1} [\tilde{\mathbf{b}}(\mathbf{z}_0) \ \tilde{\mathbf{b}}'(\mathbf{z}_0; \mathbf{M})] = \mathbf{A}_B^{-1} [\mathbf{B}\mathbf{z}_0 + \mathbf{b}_0 \ \mathbf{B}\mathbf{M}]$ and for $i \in \{1, \dots, m\}$ let $\hat{\mathbf{k}}_i$ be the i th row of \mathbf{K} . Suppose that for all $i \in \{1, \dots, m\}$, $f\text{sign}(\hat{\mathbf{k}}_i) \geq 0$. Then $\tilde{\mathbf{h}}'(\mathbf{z}_0; \mathbf{M}) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B}\mathbf{M}$, and basis B is said to be *compatible* with the directions \mathbf{M} .

Proof: Since B is optimal in the sense of Algorithm 2 in [56], $\tilde{\mathbf{h}}(\mathbf{z}_0) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \tilde{\mathbf{b}}(\mathbf{z}_0)$ and for any $\mathbf{z} \in \tilde{F}$ such that $\mathbf{A}_B^{-1} \tilde{\mathbf{b}}(\mathbf{z}) \geq \mathbf{0}$, $\tilde{\mathbf{h}}(\mathbf{z}) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \tilde{\mathbf{b}}(\mathbf{z}) = \mathbf{C}_B^T \mathbf{A}_B^{-1} (\mathbf{B}\mathbf{z} + \mathbf{b}_0)$ [56].

Notice that by assumption, $\mathbf{k}_1 \geq \mathbf{0}$. Also notice that the set $S = \{\mathbf{z} : \mathbf{A}_B^{-1} \mathbf{z} \geq \mathbf{0}\}$ is a convex set. Now consider \mathbf{k}_2 . By Proposition 7.2.1 if there exists $\delta > 0$ such that for any $\epsilon \in (0, \delta)$, $\tilde{\mathbf{h}}(\mathbf{z}_0 + \epsilon \mathbf{m}_1) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \tilde{\mathbf{b}}(\mathbf{z}_0 + \epsilon \mathbf{m}_1)$, then $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B}\mathbf{m}_1$. This will be true if for any $\epsilon \in (0, \delta)$, $\mathbf{A}_B^{-1} \tilde{\mathbf{b}}(\mathbf{z}_0 + \epsilon \mathbf{m}_1) = \mathbf{A}_B^{-1} [\mathbf{B}\mathbf{z}_0 + \mathbf{b}_0 + \epsilon \mathbf{B}\mathbf{m}_1] = \mathbf{k}_1 + \epsilon \mathbf{k}_2 \geq \mathbf{0}$. Notice that if $\mathbf{k}_1 + \delta \mathbf{k}_2 \geq \mathbf{0}$, then it will be true for any $\epsilon \in (0, \delta)$ because $\mathbf{k}_1 \geq \mathbf{0}$ and S is a convex set.

Such δ will exist if there exists $\epsilon > 0$ such that $\mathbf{k}_1 + \epsilon \mathbf{k}_2 \geq \mathbf{0}$. This can only be true if for each $i \in \{1, \dots, m\}$ such that $k_{2,i} < 0$, then $k_{1,i} > 0$. If $\mathbf{k}_2 \geq \mathbf{0}$, then set $\delta = 1$. Otherwise, if $k_{2,i} < 0$ for some i , let $\tau_i = -\frac{k_{1,i}}{k_{2,i}}$ for all i such that $k_{2,i} < 0$ and set $\delta = \min_i \tau_i$. Let $\mathbf{K}_j = [\mathbf{k}_1 \dots \mathbf{k}_j]$ and consider \mathbf{K}_2 . Let $\hat{\mathbf{k}}_{2,i}$ be the i th row

of \mathbf{K}_2 . The existence of a $\delta > 0$ is equivalent to $f\text{sign}(\widehat{\mathbf{k}}_{2,i}) \geq 0, \forall i$. Therefore, $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B} \mathbf{m}_1$. Let $\delta_1 \equiv \delta$.

Now consider $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(1)}(\mathbf{m}_2)$. From Proposition 7.2.2, $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(1)}(\mathbf{m}_2) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B} \mathbf{m}_2$ if there exists $\delta_2 > 0$ such that for any $\epsilon_2 \in (0, \delta_2)$, $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(0)}(\mathbf{m}_1 + \epsilon_2 \mathbf{m}_2) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B}[\mathbf{m}_1 + \epsilon_2 \mathbf{m}_2]$. By Proposition 7.2.1 this will require the existence of $\delta_1^* > 0$ such that for any $\epsilon_1 \in (0, \delta_1^*)$, $\tilde{\mathbf{h}}(\mathbf{z}_0 + \epsilon_1(\mathbf{m}_1 + \epsilon_2 \mathbf{m}_2)) = \mathbf{C}_B^T \mathbf{A}_B^{-1} [\mathbf{B}[\mathbf{z}_0 + \epsilon_1(\mathbf{m}_1 + \epsilon_2 \mathbf{m}_2)] + \mathbf{b}_0]$. This will be true if $\mathbf{A}_B^{-1} [\mathbf{B}[\mathbf{z}_0 + \epsilon_1(\mathbf{m}_1 + \epsilon_2 \mathbf{m}_2)] + \mathbf{b}_0] = \mathbf{k}_1 + \epsilon_1(\mathbf{k}_2 + \epsilon_2 \mathbf{k}_3) \geq \mathbf{0}$. Assume $\delta_2 > 0$ exists such that $\mathbf{k}_1 + \delta_1(\mathbf{k}_2 + \delta_2 \mathbf{k}_3) \geq \mathbf{0}$. Since $\mathbf{k}_1 \geq \mathbf{0}$ and $\mathbf{k}_1 + \delta_1 \mathbf{k}_2 \geq \mathbf{0}$, by convexity of S for all $\hat{\epsilon} \in (0, \delta_2)$, $\mathbf{k}_1 + \delta_1(\mathbf{k}_2 + \hat{\epsilon} \mathbf{k}_3) \geq \mathbf{0}$ and given a fixed $\hat{\epsilon} \in (0, \delta_2)$ for all $\epsilon^* \in (0, \delta_1)$, $\mathbf{k}_1 + \epsilon^*(\mathbf{k}_2 + \hat{\epsilon} \mathbf{k}_3) \geq \mathbf{0}$. So δ_1 is our δ_1^* and it suffices to find a $\delta_2 > 0$ such that $\mathbf{k}_1 + \delta_1(\mathbf{k}_2 + \delta_2 \mathbf{k}_3) \geq \mathbf{0}$. If $\mathbf{k}_3 \geq \mathbf{0}$, this is satisfied immediately; just set $\delta_2 = 1$. If for some i , $k_{3,i} < 0$, then we need $[\mathbf{k}_1 + \delta_1 \mathbf{k}_2]_i > 0$. For all i such that $k_{3,i} < 0$, let $\tau_i = -\frac{k_{1,i} + \delta_1 k_{2,i}}{k_{3,i}}$ and let $\delta_2 = \min_i \tau_i$. Then, $\delta_2 > 0$ if and only if $f\text{sign}(\widehat{\mathbf{k}}_{3,i}) \geq 0, \forall i$. Then, $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(1)}(\mathbf{m}_2) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B} \mathbf{m}_2$.

The proof is very similar for higher-order LD-derivatives. Consider $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1})$ with $j \in \{2, \dots, q-1\}$. From Proposition 7.2.2, $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1}) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B} \mathbf{m}_{j+1}$ if there exists $\delta_{j+1} > 0$ such that for any $\epsilon_{j+1} \in (0, \delta_{j+1})$, $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(j-1)}(\mathbf{m}_j + \epsilon_{j+1} \mathbf{m}_{j+1}) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B}[\mathbf{m}_j + \epsilon_{j+1} \mathbf{m}_{j+1}]$. Applying Proposition 7.2.2 recursively and Proposition 7.2.1, the existence of $\delta_1^* \dots, \delta_j^* > 0$ is required such that for any $\epsilon_i \in \{0, \delta_i^*\}$ with $i \in \{1, \dots, j\}$,

$$\begin{aligned} \tilde{\mathbf{h}}(\mathbf{z}_0 + \epsilon_1(\mathbf{m}_1 + \epsilon_2(\mathbf{m}_2 + \dots \epsilon_{j+1}(\mathbf{m}_{j+1})))) = \\ \mathbf{C}_B^T \mathbf{A}_B^{-1} [\mathbf{B}[\mathbf{z}_0 + \epsilon_1(\mathbf{m}_1 + \epsilon_2(\mathbf{m}_2 + \dots \epsilon_{j+1}(\mathbf{m}_{j+1})))]] + \mathbf{b}_0, \end{aligned}$$

which would require $\mathbf{k}_1 + \epsilon_1(\mathbf{k}_2 + \epsilon_2(\mathbf{k}_3 + \dots \epsilon_{j+1} \mathbf{k}_{j+2})) \geq \mathbf{0}$. Now, assume that $\mathbf{k}_1 \geq \mathbf{0}$, $\mathbf{k}_1 + \delta_1 \mathbf{k}_2 \geq \mathbf{0}, \dots$, and $\mathbf{k}_1 + \delta_1(\mathbf{k}_2 + \delta_2(\mathbf{k}_3 + \dots \delta_j(\mathbf{k}_{j+1}))) \geq \mathbf{0}$. Let $\delta_i^* = \delta_i, \forall i \in \{1, \dots, j\}$. If $\mathbf{k}_{j+2} \geq \mathbf{0}$, you can set $\delta_{j+1} = 1$ and all the requirements are satisfied from the convexity of S . If for some i , $k_{j+2,i} < 0$, find all such i and set $\tau_i = -\frac{k_{1,i} + \delta_1(k_{2,i} + \dots \delta_j k_{j+1,i})}{k_{j+2,i}}$ and set $\delta_{j+1} = \min_i \tau_i$. Consider the matrix \mathbf{K}_{j+2} and let $\widehat{\mathbf{k}}_{j+2,i}$ be the i th row of \mathbf{K}_{j+2} . Then, $\delta_{j+1} > 0$ if and only if $f\text{sign}(\widehat{\mathbf{k}}_{j+2,i}) \geq 0, \forall i$.

Then, $\tilde{\mathbf{h}}_{\mathbf{z}_0, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1}) = \mathbf{C}_B^T \mathbf{A}_B^{-1} \mathbf{B} \mathbf{m}_{j+1}$. Since for all $i \in \{1, \dots, m\}$, $f \text{sign}(\widehat{\mathbf{k}}_i) \geq 0$ (where $\widehat{\mathbf{k}}_i$ is the i th row of \mathbf{K}), the proof follows by induction. \square

Theorem 7.2.3 provides a cheaper way to evaluate LD-derivatives of \mathbf{h} at $\mathbf{u} \in \text{int}(F)$ when $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\tilde{F})$ and the available optimal LLP basis satisfies certain conditions. The requirement for $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\tilde{F})$ is not problematic, if the Phase I extension proposed in [44] and [47] is used because $\tilde{F} = \mathbb{R}^k$. However, if these conditions are not met, the LD-derivatives of \mathbf{h} must be computed using LLPs (7.11) or a different alternative approach for when numerical difficulties arise. Therefore, yet another way of finding the LD-derivatives of \mathbf{h} has been designed.

Let $\mathbf{u} \in \text{int}(F)$, $\mathbf{M} \in \mathbb{R}^{n_x \times q}$ and $j \in \{0, \dots, q-2\}$. Notice that

$$\left[[\mathbf{h}]_{\mathbf{b}, \mathbf{M}}^{(j+1)}(\mathbf{m}_{j+2}) \quad \dots \quad [\mathbf{h}]_{\mathbf{b}, \mathbf{M}}^{(q-1)}(\mathbf{m}_q) \right] = \left[[\mathbf{h}]_{\mathbf{b}, \mathbf{M}}^{(j)} \right]'(\mathbf{m}_{j+1}; \mathbf{K}_j),$$

with $\mathbf{K}_j \equiv [\mathbf{m}_{j+2} \dots \mathbf{m}_q]$. Since $[\mathbf{h}]_{\mathbf{b}, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1})$ is given by an LLP, Theorem 7.2.3 applies to calculate higher-order LD-derivatives. Algorithm 1 provides a more efficient way of solving for the LD-derivatives of \mathbf{h} .

The slightly modified Algorithm 2 can also be used to compute the LD-derivatives of \mathbf{h} and basis information. This method can be more amenable for event-detection.

Computing LD-derivatives of LLPs using optimal partition information

Consider an LP in standard form

$$\min\{\mathbf{c}^T \mathbf{x} \mid \mathbf{A} \mathbf{x} = \mathbf{b}; \mathbf{x} \geq \mathbf{0}\} \quad (7.13)$$

and its dual

$$\max\{\mathbf{b}^T \boldsymbol{\lambda} \mid \mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}\}. \quad (7.14)$$

Let X^* and Y^* be the solution sets of LPs (7.13) and (7.14), respectively. Clearly, $X^* \subset \mathbb{R}^n$ and $Y^* \subset \mathbb{R}^m$ are faces of the polyhedra describing the feasible sets of LPs

Algorithm 1 Method for finding the LD-derivatives of \mathbf{h} and basis information.

```

1: Require  $\mathbf{u} \in \text{int}(F)$ ,  $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\widetilde{F})$ , and  $\mathbf{M} \in \mathbb{R}^{n_x \times q}$ .
2: procedure CALCULATE
3:   Set  $B_0, B_1, \dots, B_q \leftarrow \emptyset$ ,  $\mathbf{K} \leftarrow []$ ,  $\widehat{\mathbf{M}} \leftarrow \mathbf{b}'(\mathbf{u}; \mathbf{M})$ ,  $\boldsymbol{\gamma} = \mathbf{0} \in \{0, 1\}^q$ .
4:   Compute  $\mathbf{h}$  and  $B_0$  using Algorithm 2 in [56].
5:   Compute  $\mathbf{K} = \mathbf{A}_{B_0}^{-1}[\mathbf{b}(\mathbf{u}) \quad \widehat{\mathbf{m}}_1]$ .
6:   For all  $i \in \{1, \dots, m\}$ , let  $l_i = \text{fsign}(\widehat{\mathbf{k}}_i)$  where  $\widehat{\mathbf{k}}_i \equiv i$ th row of  $\mathbf{K}$ .
7:   for  $j = 0 : q - 1$  do
8:      $\boldsymbol{\Lambda}^0 = \mathbf{C}_{B_0}^T \mathbf{A}_{B_0}^{-1}$ 
9:     if  $\min(\mathbf{l}) \geq 0$  then
10:       $[\mathbf{h}]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1}) = \boldsymbol{\Lambda}^j \widehat{\mathbf{m}}_{j+1}$ ,  $\boldsymbol{\Lambda}^{j+1} = \boldsymbol{\Lambda}^j$ ,  $B_{j+1} = B_j$ .
11:      if  $j < q - 1$  then
12:         $\mathbf{K} = [\mathbf{K} \quad \mathbf{A}_{B_{j+1}}^{-1}[\widehat{\mathbf{m}}_{j+2}]]$ .
13:        For all  $i$  such that  $l_i = 0$ ,  $l_i = \text{fsign}(\widehat{\mathbf{k}}_i)$ .
14:      end if
15:    else
16:       $\gamma_{j+1} = 1$ .
17:      Compute  $[\mathbf{h}]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1})$  and  $B_{j+1}$  using Algorithm 2 in [56] to solve LLP
18:      (7.11).
19:       $\boldsymbol{\Lambda}^{j+1} = [\mathbf{C}_{\mathbf{u}, \mathbf{M}}^{(j)}]_{B_{j+1}}^T [\mathbf{A}_{\mathbf{u}, \mathbf{M}}^{(j)}]_{B_{j+1}}^{-1}$ 
20:      if  $j < q - 1$  then
21:        Let  $\mathbf{K} = [\mathbf{A}_{\mathbf{u}, \mathbf{M}}^{(j)}]_{B_{j+1}}^{-1} [\widehat{\mathbf{m}}_{j+1} \quad \widehat{\mathbf{m}}_{j+2}]$  and recompute  $\widehat{\mathbf{k}}_i$  for all  $i$ .
22:        For all  $i$ ,  $l_i = \text{fsign}(\widehat{\mathbf{k}}_i)$ .
23:      end if
24:    end for
25: return  $\mathbf{h}'(\mathbf{u}; \mathbf{M})$ ,  $\boldsymbol{\gamma}$ ,  $\boldsymbol{\Lambda}^k$ ,  $B_0, \dots, B_q$  for all  $k \in 0, \dots, q$ .
26: end procedure

```

(7.13) and (7.14), respectively. The optimal partition \widehat{B}, \widehat{N} is defined as in [120, 1]:

$$\widehat{B} \equiv \{j | x_j > 0 \text{ for some } \mathbf{x} \in X^* \text{ and } j = 1, \dots, n\},$$

$$\widehat{N} \equiv \{j | c_j - [\mathbf{a}_j]^T \boldsymbol{\lambda} > 0 \text{ for some } \boldsymbol{\lambda} \in Y^* \text{ and } j = 1, \dots, n\}.$$

Algorithm 2 Modified method for finding the LD-derivatives of \mathbf{h} and basis information.

- 1: Require $\mathbf{u} \in \text{int}(F)$, $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\widetilde{F})$, and $\mathbf{M} \in \mathbb{R}^{n_x \times q}$.
 - 2: **procedure** CALCULATE
 - 3: Set $B_0, B_1, \dots, B_q \leftarrow \emptyset$, $\mathbf{K} \leftarrow []$, $\widehat{\mathbf{M}} \leftarrow \mathbf{b}'(\mathbf{u}; \mathbf{M})$, $\gamma = 0$, $j = 0$.
 - 4: Compute \mathbf{h} and B_0 using Algorithm 2 in [56].
 - 5: Compute the optimal dual vertices $\boldsymbol{\lambda}_{i,j}$ for $i \in \{0, \dots, n_h\}$.
 - 6: $\boldsymbol{\Lambda}^0 = \mathbf{C}_{B_0}^T \mathbf{A}_{B_0}^{-1}$.
 - 7: Compute $\mathbf{K} = \mathbf{A}_{B_0}^{-1} [\mathbf{b}(\mathbf{u}) \quad \widehat{\mathbf{M}}]$.
 - 8: For all $i \in \{1, \dots, m\}$, let $l_i = \text{fsign}(\widehat{\mathbf{k}}_i)$ where $\widehat{\mathbf{k}}_i \equiv i$ th row of \mathbf{K} .
 - 9: **while** $\min(\mathbf{l}) < 0$ **do**
 - 10: Compute $[\mathbf{h}]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1})$ and B_{j+1} using Algorithm 2 in [56] to solve LLP (7.11).
 - 11: $\boldsymbol{\Lambda}^{j+1} = [\mathbf{C}_{\mathbf{u}, \mathbf{M}}^{(j)}]_{B_{j+1}}^T [\mathbf{A}_{\mathbf{u}, \mathbf{M}}^{(j)}]_{B_{j+1}}^{-1}$.
 - 12: Let $\mathbf{K} = [\mathbf{A}_{\mathbf{u}, \mathbf{M}}^{(j)}]_{B_{j+1}}^{-1} [\widehat{\mathbf{m}}_{j+1} \quad \dots \quad \widehat{\mathbf{m}}_q]$ and recompute $\widehat{\mathbf{k}}_i$ for all i .
 - 13: For all i , $l_i = \text{fsign}(\widehat{\mathbf{k}}_i)$.
 - 14: $j = j + 1$.
 - 15: **end while**
 - 16: $\gamma = j$.
 - 17: **for** $k = j : q - 1$ **do**
 - 18: $[\mathbf{h}]_{\mathbf{u}, \mathbf{M}}^{(k)}(\mathbf{m}_{k+1}) = \boldsymbol{\Lambda}^j \widehat{\mathbf{m}}_{k+1}$.
 - 19: $B_{k+1} = B_j$.
 - 20: **end for**
 - 21: **return** $\mathbf{h}'(\mathbf{u}; \mathbf{M})$, γ , $\boldsymbol{\Lambda}^k$, B_0, \dots, B_q for all $k \in \{0, \dots, q\}$ and $\boldsymbol{\lambda}_{i,j}$ for $i \in \{0, \dots, n_h\}$ and $j \in \{1, \dots, \gamma\}$.
 - 22: **end procedure**
-

From Proposition 2.2 in [1], if X^* is nonempty, then $\widehat{B} \cap \widehat{N} = \emptyset$ and $\widehat{B} \cup \widehat{N} = \{1, \dots, n\}$.

Then,

$$\begin{aligned}
 X^* &= \{\mathbf{x} \mid \mathbf{A}_{\widehat{B}} \mathbf{x}_{\widehat{B}} = \mathbf{b}; \mathbf{x}_{\widehat{B}} \geq \mathbf{0}, \mathbf{x}_{\widehat{N}} = \mathbf{0}\}, \\
 Y^* &= \{\boldsymbol{\lambda} \mid \mathbf{A}_{\widehat{B}}^T \boldsymbol{\lambda} = \mathbf{c}_{\widehat{B}}; \mathbf{A}_{\widehat{N}}^T \boldsymbol{\lambda} \leq \mathbf{c}_{\widehat{N}}\}.
 \end{aligned} \tag{7.15}$$

Computation of the sets \widehat{B}, \widehat{N} can be difficult. One way is to obtain a strictly complementary solution, which can be found in the relative interior of X^* . Then, the optimal partition is given by this solution point (Theorem 2.1 in [49]). Another way of obtaining the partition is by traversing the optimal basic solutions, but this can be computationally expensive ([49]). Here, we use lexicographic optimization. Subsets of

\widehat{B} and \widehat{N} can be computed with the existing optimal basic solution by finding strictly positive variables and reduced costs, respectively. All remaining variables (those being equal to zero and with reduced cost equal to zero) can be maximized one by one while adding the constraint that the objective needs to be satisfied. If the objective of this LP is greater than zero, then that variable corresponds to \widehat{B} ; otherwise, it corresponds to \widehat{N} . This process can be done more efficiently by maximizing several variables simultaneously.

Consider LLP (7.4) with $\mathbf{u} \in \text{int}(F)$ and $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\widetilde{F})$, and let $\widehat{B}^i(\mathbf{u}), \widehat{N}^i(\mathbf{u})$ be the optimal partition sets and Y_i^* the dual solution set for each level of LLP (7.4) with $i \in \{0, \dots, n_h\}$. Notice that $\widehat{B}^{i+1}(\mathbf{u}) \subset \widehat{B}^i(\mathbf{u})$ for $i \in \{0, \dots, n_h - 1\}$. Consider $\mathbf{d} \in \mathbb{R}^{n_x}$. Then, the directional derivative $\mathbf{h}'(\mathbf{u}; \mathbf{d})$ is given by LLP (7.11) or LPs (7.9). For $\mathbf{u} \in \text{int}(F)$, $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\widetilde{F})$, $\mathbf{M} \in \mathbb{R}^{n_x \times q}$, and $i \in \{0, \dots, n_h\}$, LPs (7.9) result in the following LLP:

$$\begin{aligned} \left[h_i(\mathbf{u}) \quad h'_i(\mathbf{u}; \mathbf{M}) \right] &= \text{lex} \max_{\boldsymbol{\lambda} \in \mathbb{R}^{m+i}} [\mathbf{Q}^i(\mathbf{u}, \mathbf{M})]^\top \boldsymbol{\lambda}, \\ &\text{s.t. } [\mathbf{A}^i]^\top \boldsymbol{\lambda} \leq \mathbf{c}_i, \end{aligned} \quad (7.16)$$

where $\mathbf{Q}^i(\mathbf{u}, \mathbf{M}) \equiv \left[\mathbf{q}^i(\mathbf{u}) \quad [\mathbf{q}^i]'(\mathbf{u}; \mathbf{M}) \right]$. Let $i \in \{0, \dots, n_h\}$ and consider

$$h'_i(\mathbf{u}; \mathbf{d}) = \max\{[\mathbf{q}^i]'(\mathbf{u}; \mathbf{d})^\top \boldsymbol{\lambda} \mid \boldsymbol{\lambda} \in Y_i^*\}. \quad (7.17)$$

which result in the following equivalent LPs:

$$\begin{aligned} h'_i(\mathbf{u}; \mathbf{d}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^{m+i}} [\mathbf{q}^i]'(\mathbf{u}; \mathbf{d})^\top \boldsymbol{\lambda}, \\ &\text{s.t.} \quad [\mathbf{A}^i]^\top \boldsymbol{\lambda} \leq \mathbf{c}_i, \\ &\quad -\mathbf{q}^i(\mathbf{u})^\top \boldsymbol{\lambda} \leq -h_i(\mathbf{u}), \end{aligned}$$

and

$$\begin{aligned}
h'_i(\mathbf{u}; \mathbf{d}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^{m+i}} [\mathbf{q}^i]'(\mathbf{u}; \mathbf{d})^T \boldsymbol{\lambda}, & (7.18) \\
\text{s.t.} \quad & [\mathbf{A}^i_{\widehat{B}^i(\mathbf{u})}]^T \boldsymbol{\lambda} = [\mathbf{c}_i]_{\widehat{B}^i(\mathbf{u})}, \\
& [\mathbf{A}^i_{\widehat{N}^i(\mathbf{u})}]^T \boldsymbol{\lambda} \leq [\mathbf{c}_i]_{\widehat{N}^i(\mathbf{u})}.
\end{aligned}$$

If all $i \in \{0, \dots, n_h\}$ are considered simultaneously, the duals of LPs in (7.18) do not result in a LLP (compared to LLP (7.11)), because the sets $\widehat{B}^i(\mathbf{u})$, $\widehat{N}^i(\mathbf{u})$ are different for each i . For fixed i the LPs in (7.9) result in LLP (7.16) with q levels. Based on these observations, Algorithm 3 computes the LD-derivatives of an LLP in the cases Algorithm 1 fails due to numerical difficulties.

Consider the dual of LP (7.18):

$$\begin{aligned}
[h_i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{d}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_i^T \mathbf{v}, & (7.19) \\
\text{s.t.} \quad & \mathbf{A}^i \mathbf{v} = [\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{d}), \\
& \mathbf{v}_{\widehat{N}^i(\mathbf{u})} \geq \mathbf{0}.
\end{aligned}$$

LP (7.19) can be converted into standard form. Let $\mathbf{A}^{i,0} \equiv \mathbf{A}^i$, $\mathbf{A}^{i,1} \equiv \begin{bmatrix} \mathbf{A}^i & -\mathbf{A}^i_{\widehat{B}^i(\mathbf{u})} \end{bmatrix}$, $\mathbf{c}_i^0 \equiv \mathbf{c}_i$, $\mathbf{c}_i^1 \equiv \begin{bmatrix} \mathbf{c}_i \\ -[\mathbf{c}_i]_{\widehat{B}^i(\mathbf{u})} \end{bmatrix}$ and $\gamma_i^{(0)} = \text{card}(\widehat{B}^i(\mathbf{u}))$. Then,

$$\begin{aligned}
[h_i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{d}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v + \gamma_i^{(0)}}} [\mathbf{c}_i^1]^T \mathbf{v}, & (7.20) \\
\text{s.t.} \quad & \mathbf{A}^{i,1} \mathbf{v} = [\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(0)}(\mathbf{d}), \\
& \mathbf{v} \geq \mathbf{0}.
\end{aligned}$$

Let $\widehat{B}^{i,0}(\mathbf{u}, \mathbf{M}) \equiv \widehat{B}^i(\mathbf{u})$, $\widehat{N}^{i,0}(\mathbf{u}, \mathbf{M}) \equiv \widehat{N}^i(\mathbf{u})$ and let $\widehat{B}^{i,1}(\mathbf{u}, \mathbf{M})$, $\widehat{N}^{i,1}(\mathbf{u}, \mathbf{M})$ refer to the optimal partition of LP (7.20). For $j \in \{2, \dots, q\}$, let $\mathbf{A}^{i,j} \equiv \begin{bmatrix} \mathbf{A}^{i,j-1} & -\mathbf{A}^{i,j-1}_{\widehat{B}^{i,j}(\mathbf{u}, \mathbf{M})} \end{bmatrix}$, $\mathbf{c}_i^j \equiv \begin{bmatrix} \mathbf{c}_i^{j-1} \\ -[\mathbf{c}_i^{j-1}]_{\widehat{B}^{i,j}(\mathbf{u}, \mathbf{M})} \end{bmatrix}$ and $\gamma_i^{(j)} = \gamma_i^{(j-1)} + \text{card}(\widehat{B}^{i,j}(\mathbf{u}, \mathbf{M}))$. Then $[h_i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d})$ is given

by LP

$$\begin{aligned}
[h_i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v + \gamma_i^{(j)}}} [\mathbf{c}_i^{j+1}]^T \mathbf{v}, \\
\text{s.t. } \mathbf{A}^{i,j+1} \mathbf{v} &= [\mathbf{q}^i]_{\mathbf{u}, \mathbf{M}}^{(j)}(\mathbf{d}), \\
\mathbf{v} &\geq \mathbf{0}.
\end{aligned} \tag{7.21}$$

Algorithm 3 Method for finding the LD-derivatives of \mathbf{h} using the optimal partition.

- 1: Require $\mathbf{u} \in \text{int}(F)$, $\widehat{\mathbf{b}}(\mathbf{u}) \in \text{int}(\tilde{F})$, and $\mathbf{M} \in \mathbb{R}^{n_x \times q}$.
 - 2: **procedure** CALCULATE
 - 3: Set $\widehat{\mathbf{M}} \leftarrow \mathbf{b}'(\mathbf{u}; \mathbf{M})$, $\gamma = \mathbf{0}_{n_h+1}$.
 - 4: Solve LLP (7.4) using Algorithm 2 in [56].
 - 5: Compute the optimal bases $B_{i,0}$ for each level $i \in \{0, \dots, n_h\}$.
 - 6: $\boldsymbol{\lambda}^{i,0} = [\mathbf{c}_i^0]_{B_{i,0}}^T [\mathbf{A}_{B_{i,0}}^{i,0}]^{-1}$ for all $i \in \{0, \dots, n_h\}$.
 - 7: **for** $i = 0 : n_h$ **do**
 - 8: **for** $j = 0 : q - 1$ **do**
 - 9: Let $\mathbf{K} = [\mathbf{A}_{B_{i,0}}^{i,0}]^{-1} [\widehat{\mathbf{m}}_{j+1} \dots \widehat{\mathbf{m}}_q]$.
 - 10: For all k , $l_k = \text{sign}(\widehat{\mathbf{k}}_k)$, where $\widehat{\mathbf{k}}_k \equiv k$ th row of \mathbf{K} .
 - 11: **if** $\min(\mathbf{l}) \geq 0$ **then**
 - 12: For all $k \in \{j, \dots, q - 1\}$, $[h_i]_{\mathbf{u}, \mathbf{M}}^{(k)}(\mathbf{m}_{k+1}) = \mathbf{c}_{i, B_{i,j}}^T [\mathbf{A}_{B_{i,j}}^i]^{-1} \widehat{\mathbf{m}}_{k+1}$.
 - 13: For all $k \in \{j + 1, \dots, q\}$, $\boldsymbol{\lambda}^{i,k} = \boldsymbol{\lambda}^{i,j}$.
 - 14: $\gamma_i = j$.
 - 15: Break for loop.
 - 16: **end if**
 - 17: Compute the optimal partition sets $\widehat{B}^{i,j}(\mathbf{u})$, $\widehat{N}^{i,j}(\mathbf{u})$.
 - 18: Solve LP (7.21) using dual simplex and warm-starting with basis $B_{i,j}$.
 - 19: Find $[h_i]_{\mathbf{b}, \mathbf{M}}^{(j)}(\mathbf{m}_{j+1})$ and find an optimal basis $B_{i,j+1}$.
 - 20: $\boldsymbol{\lambda}^{i,j+1} = [\mathbf{c}_i^{j+1}]_{B_{i,j+1}}^T [\mathbf{A}_{B_{i,j+1}}^{i,j+1}]^{-1}$
 - 21: **end for**
 - 22: $\widehat{\mathbf{M}} = \begin{bmatrix} \widehat{\mathbf{M}} \\ [h_i]'(\mathbf{u}; \mathbf{M}) \end{bmatrix}$.
 - 23: **end for**
 - 24: **return** $\mathbf{h}'(\mathbf{u}; \mathbf{M})$, $\boldsymbol{\lambda}^{i,j}$, $\mathbf{A}_{B_{i,j}}^{i,j}$ for all $i \in \{0, \dots, n_h\}$ and $j \in \{0, \dots, q\}$, γ .
 - 25: **end procedure**
-

7.3 Efficient integration of ODE (7.6) to obtain the LD-derivatives of ODE systems with LLPs embedded

ODE systems with embedded LLPs parameterized by their right-hand sides can be transformed into DAE systems [56]. At a first glance, it seems that the same reasoning cannot be applied to the embedded LLPs in (7.6), as they are parameterized by their right hand side and some columns in their technology matrix (see LLP (7.11)). A naive approach would be to apply an approach similar to the one in [56, 59], find a basis and track both primal and dual feasibility. The following example illustrates the difficulties of obtaining sensitivity information of ODE systems with LLPs embedded and how tracking primal and dual feasibility of the bases associated to sensitivity information can lead to numerical singularities. The embedded LP in the following example is small enough to find its parametric solution, but for many applications, solving the LP parametrically is an intractable approach.

Example 7.3.1. Consider the following ODE system with a LP embedded:

$$\begin{aligned} \frac{dy_1}{dt}(t, \mathbf{p}) &= 0, \frac{dy_2}{dt}(t, \mathbf{p}) = e^{-h(\mathbf{y}(t, \mathbf{p}))}, \forall t \in (0, 15), \mathbf{y}(0, \mathbf{p}) = \mathbf{p}, \\ h(\mathbf{z}) &= \min_{\mathbf{v} \in \mathbb{R}^4} \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \mathbf{v} \\ \text{s.t.} \quad & \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \mathbf{v} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{7.22}$$

The dual of the embedded LP is

$$\begin{aligned} h(\mathbf{z}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^2} \mathbf{z}^T \boldsymbol{\lambda} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\lambda} \leq \mathbf{1}. \end{aligned} \tag{7.23}$$

Therefore,

$$h(\mathbf{z}) = \begin{cases} z_1 & \text{if } z_1 \geq 0, z_2 < 0, \\ z_1 + z_2, & \text{if } z_1 \geq 0, z_2 \geq 0, \\ z_2 & \text{if } z_2 \geq 0, z_1 < 0, \\ 0 & \text{if } \mathbf{z} < \mathbf{0}. \end{cases}$$

Let $\mathbf{p}_0 = (1, -1)$. Then

$$\mathbf{y}(t, \mathbf{p}_0) = (1, te^{-1} - 1), \quad \forall t \in (0, e],$$

$$\mathbf{y}(t, \mathbf{p}_0) = (1, \ln(t) - 1), \quad \forall t \in (e, 15).$$

The directional derivative of h is given by:

$$\begin{aligned} h'(\mathbf{z}; \mathbf{d}) &= \min_{\mathbf{v} \in \mathbb{R}^5} \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & -h(\mathbf{z}) \end{array} \right] \mathbf{v} & (7.24) \\ \text{s.t.} & \left[\begin{array}{ccccc} 1 & 0 & -1 & 0 & -z_1 \\ 0 & 1 & 0 & -1 & -z_2 \end{array} \right] \mathbf{v} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned}$$

Let $\mathbf{y}_t(\mathbf{p}) = \mathbf{y}(t, \mathbf{p})$. Then, $\mathbf{y}'_t(\mathbf{p}; \mathbf{d})$ is given by the solution of the following ODE:

$$\frac{d\mathbf{s}}{dt}(t) = \begin{bmatrix} 0 \\ -e^{-h(\mathbf{y}(t, \mathbf{p}))} h'(\mathbf{y}(t, \mathbf{p}); \mathbf{s}(t)) \end{bmatrix}, \quad \mathbf{s}(t_0) = \mathbf{d}. \quad (7.25)$$

Let $\mathbf{d} = (0, 1)$. Then $s_1(t) = 0$ for all time and $h'(\mathbf{y}(t, \mathbf{p}_0); \mathbf{s}(t)) = 0$ when $y_2(t, \mathbf{p}_0) < 0$ and $h'(\mathbf{y}(t, \mathbf{p}_0); \mathbf{s}(t)) = s_2(t)$ when $y_2(t, \mathbf{p}_0) \geq 0$. Therefore, the ODE system (7.25) presents a discontinuity in its right-hand side introduced by the solution of LP (7.24). This discontinuity is associated to a change in the optimal basis. Any numerical integrator without event detection will take a very large number of steps near the discontinuity. Therefore, a way to detect this discontinuity will make numerical integration faster.

The optimal basis of LP (7.24) at $t = 0$ is $B = \{1, 5\}$ and the basis matrix is

$$\mathbf{B}(t) = \begin{bmatrix} 1 & -y_1(t, \mathbf{p}_0) \\ 0 & -y_2(t, \mathbf{p}_0) \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 - te^{-1} \end{bmatrix}.$$

For this simple example we know that given \mathbf{p}_0 and \mathbf{d} , this basis will remain optimal as long as $y_2(t, \mathbf{p}_0) < 0$. If the example were not this simple, we would need to verify instead that the following three conditions were satisfied:

1. $\mathbf{B}(t)$ is of maximal rank,
2. Primal feasibility given by $(\mathbf{B}(t))^{-1}\mathbf{z}(t) \geq \mathbf{0}$,
3. Dual feasibility given by $\mathbf{c}(t) - \mathbf{A}(t)^T(\mathbf{B}(t)^T)^{-1}\mathbf{c}_B(t) \geq \mathbf{0}$,

where $\mathbf{A}(t)$ and $\mathbf{c}(t)$ refer to the technology matrix and the cost vector of LP (7.24) at time t .

However,

$$\lim_{t \rightarrow e} \mathbf{B}(t) = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix},$$

therefore, as $t \rightarrow e$, neither primal nor dual feasibility can be verified reliably. In addition, $\mathbf{B}(t)$ is maximal rank unless $t = e$, therefore, this basis matrix is optimal despite approaching a point of singularity. Therefore, in this case tracking primal and dual feasibility of the basis matrix associated to the directional derivative does not work.

It is important to notice that the sensitivity system can present state-dependent discontinuities, as reported in [72]. This issue is not resolved in [60]. The following section describes how to integrate ODE (7.25) efficiently and without encountering singular basis matrices.

7.3.1 Reformulation of ODE (7.6) into a DAE system

Consider ODE (7.1). If $\mathbf{z} \in \text{int}(F)$, then from strong duality of LPs [12] and for $i \in \{0, \dots, n_h\}$:

$$\begin{aligned} h_i(\mathbf{z}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^{m+i}} \mathbf{q}^i(\mathbf{z})^\top \boldsymbol{\lambda} \\ \text{s.t. } & [\mathbf{A}^i]^\top \boldsymbol{\lambda} \leq \mathbf{c}_i. \end{aligned} \quad (7.26)$$

Definition 7.3.1. Consider ODE (7.1). Let DE_i be the set of extreme points and $DO_i(t)$ contain the optimal extreme points of LP (7.26) evaluated at $\mathbf{x}(t, \mathbf{p})$. In case $n_h = 0$, refer to these sets as DE and $DO(t)$, respectively.

Under Assumption 7.2.1 and since \mathbf{A}^i is full row rank, Theorem 2.6 in [12] implies that the sets DE_i are nonempty and finite. Since the feasible sets of these LPs remain constant, the sets DE_i remain constant too. Since LP mappings are abs-factorable (Lemma A.3 in [73]), ODE (7.1) is an ODE system with an abs-factorable right-hand side and can be integrated using an event-detection scheme. This event-detection scheme relies on finding when an argument to any absolute value function in the factored representation of the right-hand side of the ODE system crosses zero, and the number of such events is finite (Theorem 3.12 in [73]). Given that LPs mappings are abs-factorable, we can present the following definition.

Definition 7.3.2. Consider ODE (7.1). For appropriate $\delta > 0$, $t^* \in (t_0, t_f)$, and for all $i \in \{0, \dots, n_h\}$, let $DO_i^L(t^*)$ be the constant value of $DO_i(t)$ on $t \in (t^* - \delta, t^*)$ and $DO_i^R(t^*)$ be the constant value of $DO_i(t)$ on $t \in (t^*, t^* + \delta)$. In the case of $n_h = 0$, refer to this sets simply as $DO^L(t^*)$ and $DO^R(t^*)$, respectively.

Assumption 7.3.1. Consider ODE (7.1) and let Assumption 7.2.1 hold. Assume that $\mathbf{x}(t, \mathbf{p}) \in F$ for all $t \in [t_0, t_f]$.

Assumption 7.3.1 implies that the sets $DO_i(t)$ are nonempty for all $t \in [t_0, t_f]$ and for all $i \in \{0, \dots, n_h\}$.

Lemma 7.3.1. Consider ODE (7.1) with $n_h = 0$ (LP instead of LLP) and let Assumption 7.3.1 hold. Let $t_j, t_k \in [t_0, t_f]$ and $[t_j, t_k] \cap Z_t = \widehat{t}$, where Z_t is the finite set that contains points at which $DO^L(t) \neq DO^R(t)$ (see Corollary A.5 in [73]). Then $DO^L(\widehat{t}) \cup DO^R(\widehat{t}) \subset DO(\widehat{t})$.

Proof: Assume this is not true. Without loss of generality (the following applies to $DO^R(\widehat{t})$ too), there exists $\boldsymbol{\lambda}^* \in DO^L(\widehat{t}), \boldsymbol{\lambda}^* \notin DO(\widehat{t})$. We know that $DO(t) = \arg \max_{\boldsymbol{\lambda} \in DE} \mathbf{q}^0(\mathbf{x}(t, \mathbf{p}_0))^T \boldsymbol{\lambda}$. Since \mathbf{x} is the solution to an ODE system, it is continuous, and since \mathbf{q}^0 and h_0 are continuous functions, so are the compositions $\mathbf{q}^0 \circ \mathbf{x}$ and $h_0 \circ \mathbf{x}$. For appropriate $\epsilon > 0$ and for all $t \in (\widehat{t} - \epsilon, \widehat{t})$, $h_0(\mathbf{x}(t, \mathbf{p}_0)) = \mathbf{q}^0(\mathbf{x}(t, \mathbf{p}_0))^T \boldsymbol{\lambda}^*$, but since $\boldsymbol{\lambda}^* \notin DO(\widehat{t})$, $h_0(\mathbf{x}(\widehat{t}, \mathbf{p}_0)) \neq \mathbf{q}^0(\mathbf{x}(\widehat{t}, \mathbf{p}_0))^T \boldsymbol{\lambda}^*$. From continuity, $h_0(\mathbf{x}(\widehat{t}, \mathbf{p}_0)) = \lim_{\tau \rightarrow 0^+} h_0(\mathbf{x}(\widehat{t} - \tau, \mathbf{p}_0)) = \lim_{\tau \rightarrow 0^+} \mathbf{q}^0(\mathbf{x}(\widehat{t} - \tau, \mathbf{p}_0))^T \boldsymbol{\lambda}^*$. This means $\lim_{\tau \rightarrow 0^+} \mathbf{q}^0(\mathbf{x}(\widehat{t} - \tau, \mathbf{p}_0))^T \boldsymbol{\lambda}^* \neq \mathbf{q}^0(\mathbf{x}(\widehat{t}, \mathbf{p}_0))^T \boldsymbol{\lambda}^*$ which contradicts the continuity of the vector product function. \square

Remark 7.3.1. Consider ODE (7.1) with $n_h > 0$ (embedded LLP) and let Assumption 7.3.1 hold. Let $t_j, t_k \in [t_0, t_f]$ and $[t_j, t_k] \cap Z_t = \widehat{t}$, where Z_t is the finite set that contains points at which $DO_i^L(t) \neq DO_i^R(t)$ for any $i \in \{0, \dots, n_h\}$ (see Corollary A.5 in [73]). Since \mathbf{q}^i and h_i are continuous for all $i \in \{0, \dots, n_h\}$, Lemma 7.3.1 implies $DO_i^L(\widehat{t}) \cup DO_i^R(\widehat{t}) \subset DO_i(\widehat{t})$ for all i .

Corollary A.5 in [73] shows that for a time domain $[t_0, t_f]$, there exists a finite ordered set $Z_t \subset [t_0, t_f]$ such that for $t_{j-1}, t_j \in Z_t, t_{j-1} < t_j, Z_t \cap (t_{j-1}, t_j) = \emptyset$, such that for $t^* \in [t_0, t_f], t^* \notin Z_t, DO_i^L(t^*) = DO_i^R(t^*)$ for all $i \in \{0, \dots, n_h\}$. Essentially given $t_{j-1}, t_j \in Z_t, DO_i(t)$ is constant almost everywhere on (t_{j-1}, t_j) for all i . Given ODE (7.1), we are interested in detecting when a $t \in Z_t$ has been crossed.

Theorem 7.3.2. Consider ODE (7.1) with $n_h = 0$ (LP instead of LLP) and let Assumption 7.3.1 hold. Let $t_j, t_k \in [t_0, t_f]$ and $[t_j, t_k] \cap Z_t = \widehat{t}$, where Z_t is the finite set that contains points at which $DO^L(t) \neq DO^R(t)$ (see Corollary A.5 in [73]). If no argument to an absolute value function crosses zero in an abs-factorable representation of \mathbf{f} or \mathbf{b} at \widehat{t} , $DO^L(\widehat{t}) \cap DO^R(\widehat{t}) = \emptyset$.

Proof: According to Lemma A.3 in [73], LP-mappings are abs-factorable and can be

factored out as a composition of absolute value functions. There are many different ways of obtaining an abs-factorable representation of a particular LP. Specifically, let DE be the set of dual extreme points of the embedded LP. This set is constant in the case of LPs parameterized by their right-hand side. Let DE contain α elements and for each element $\mathbf{p} \in DE$, the following factored representation can be constructed:

$$\begin{aligned}
h(\mathbf{z}) &= \max \left\{ \max \left\{ \mathbf{p}^T \mathbf{q}^0(\mathbf{z}), \mathbf{k}^T \mathbf{q}^0(\mathbf{z}) \right\}, \forall \mathbf{k} \in DE \right\} & (7.27) \\
&= \max \left\{ \frac{1}{2}(\mathbf{p} + \mathbf{k})^T \mathbf{q}^0(\mathbf{z}) + \frac{1}{2} |(\mathbf{p} - \mathbf{k})^T \mathbf{q}^0(\mathbf{z})|, \forall \mathbf{k} \in DE \right\} \\
&= \max \{ \theta_1, \theta_2, \dots, \theta_{\alpha-1} \}, \\
\theta_j &\equiv \frac{1}{2}(\mathbf{p} + \mathbf{k}_{(j)})^T \mathbf{q}^0(\mathbf{z}) + \frac{1}{2} |(\mathbf{p} - \mathbf{k}_{(j)})^T \mathbf{q}^0(\mathbf{z})|, \text{ for } j \in \{1, \dots, \alpha - 1\},
\end{aligned}$$

which can further be expanded into more absolute value functions. Without loss of generality, let us consider the following expansion of the max operator to absolute value functions:

$$h(\mathbf{z}) = \max \dots \max(\max(\theta_1, \theta_2), \theta_3), \dots, \theta_{\alpha-1}). \quad (7.28)$$

Each $\max(a, b)$ operator can be expressed as $\frac{1}{2}(a + b) + \frac{1}{2}|a - b|$. If $\mathbf{p} \in DO$, $\theta_1 = \theta_2 = \dots = \theta_{\alpha-1}$ and they are all equal to $h(\mathbf{z})$. In this case, all arguments to absolute value functions in the expansion of (7.28) will be equal to zero.

Let us assume that for appropriate $\epsilon > 0$ and any $t \in (\hat{t} - \epsilon, \hat{t}) \equiv T_L$, $DO(t) = DO^L(\hat{t})$ and contains $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_\kappa$ with $\kappa \leq \alpha$, and $h(\mathbf{x}(t, \mathbf{p}_0)) = (\mathbf{p}_1)^T (\mathbf{q}^0(\mathbf{x}(t, \mathbf{p}_0))) = \beta(t)$. Let $\mathbf{z}_t \equiv \mathbf{x}(t, \mathbf{p}_0)$ and $t_1 \in T_L$. Without loss of generality, take the factored representation corresponding to \mathbf{p}_1 :

$$\begin{aligned}
h(\mathbf{z}_{t_1}) &= \max \left\{ \frac{1}{2}(\mathbf{p}_1 + \mathbf{k})^T \mathbf{q}^0(\mathbf{z}_{t_1}) + \frac{1}{2} |(\mathbf{p}_1 - \mathbf{k})^T \mathbf{q}^0(\mathbf{z}_{t_1})|, \forall \mathbf{k} \in DE \right\} & (7.29) \\
&= \max \{ \beta(t_1), \dots, \beta(t_1) \}.
\end{aligned}$$

If this abs-factorable representation of the embedded LP is used, the following statements are true:

1. $(\mathbf{p}_1 - \mathbf{k})^T \mathbf{q}^0(\mathbf{z}_{t_1}) = 0$ for all $\mathbf{k} \in DO(t_1) = DO^L(\hat{t})$,
2. $(\mathbf{p}_1 - \mathbf{k})^T \mathbf{q}^0(\mathbf{z}_{t_1}) > 0$ for all $\mathbf{k} \in DE \setminus DO^L(\hat{t})$,
3. All absolute value functions in the expansion (7.28) are equal to zero.

These statements remain true for any $t_1 \in T_L$ because DO is constant on this interval by assumption. For appropriate $\epsilon > 0$ and $t_2 \in (\hat{t}, \hat{t} + \epsilon) \equiv T_R$, $DO(t_2) = DO^R(\hat{t}) \neq DO^L(\hat{t})$ by assumption. Assume without loss of generality that there exists an extreme point \mathbf{p}^* which is present in both $DO^L(\hat{t})$ and $DO^R(\hat{t})$ and consider the abs-factorable representation of h associated to \mathbf{p}^* . If this abs-factorable representation is used, all absolute value functions in the expansion (7.28) remain equal to zero and all θ_i remain equal to $h(\mathbf{z}_t)$. For DO to change, at least one of the following two statements must be true:

1. $\mathbf{k}^* \in DO^L(\hat{t})$, $\mathbf{k}^* \notin DO^R(\hat{t})$. Then, $(\mathbf{p}^* - \mathbf{k}^*)^T \mathbf{q}^0(\mathbf{z}_{t_1}) = 0$ but $(\mathbf{p}^* - \mathbf{k}^*)^T \mathbf{q}^0(\mathbf{z}_{t_2}) > 0$.
2. $\mathbf{k}^* \notin DO^L(\hat{t})$, $\mathbf{k}^* \in DO^R(\hat{t})$. Then, $(\mathbf{p}^* - \mathbf{k}^*)^T \mathbf{q}^0(\mathbf{z}_{t_1}) > 0$ but $(\mathbf{p}^* - \mathbf{k}^*)^T \mathbf{q}^0(\mathbf{z}_{t_2}) = 0$.

According to Corollary 3.15 in [73], for any of these cases to occur, an argument to an absolute value function has to cross zero at time \hat{t} in the abs-factorable representations of \mathbf{f} , \mathbf{b} , or h . By assumption, there is no argument to an absolute value function crossing zero in an abs-factorable representation of \mathbf{f} and \mathbf{b} . In addition, given this abs-factorable representation of h associated to \mathbf{p}^* , no argument to an absolute value function would cross zero. Then, a contradiction is reached and $DO^L(\hat{t}) \cap DO^R(\hat{t}) = \emptyset$.

Since $DO(t)$ is nonempty for all $t \in [t_j, t_k]$ by Assumption 7.3.1, the only way $DO^L(\hat{t}) \neq DO^R(\hat{t})$ is if some $\mathbf{k}^* \notin DO^L(\hat{t})$ is in $DO^R(\hat{t})$ and all $\mathbf{p}^* \in DO^L(\hat{t})$ are not in $DO^R(\hat{t})$. In that case $(\mathbf{p}^* - \mathbf{k}^*)^T \mathbf{q}^0(\mathbf{z}_{t_1}) > 0$ and $(\mathbf{p}^* - \mathbf{k}^*)^T \mathbf{q}^0(\mathbf{z}_{t_2}) < 0$ for $t_1 \in T_L$ and $t_2 \in T_R$, which is consistent with Corollary 3.15 in [73]. \square

Theorem 7.3.2 implies that when the right-hand side of LP (7.2) is parameterized by an analytic function of a single variable, when $DO^L(t^*) \neq DO^R(t^*)$ then $DO^L(t^*) \cap$

$DO^R(t^*) = \emptyset$. A similar result can be found in Corollary 3.2 in [1], when the right-hand side of LP (7.2) is parameterized by an affine function of a single variable. In this sense, Theorem 7.3.2 is more general.

Corollary 7.3.1. Consider ODE (7.1) with $n_h > 0$ (embedded LLP) and let Assumption 7.3.1 hold. These assumptions imply $DO_i(t)$ are nonempty for all $t \in [t_0, t_f]$ and for all $i \in \{0, \dots, n_h\}$. Let $t_j, t_k \in [t_0, t_f]$ and $[t_j, t_k] \cap Z_t = \hat{t}$, where Z_t is the finite set that contains points at which $DO_i^L(t) \neq DO_i^R(t)$ for any $i \in \{0, \dots, n_h\}$ (see Corollary A.5 in [73]). If no argument to an absolute value function crosses zero in an abs-factorable representation of \mathbf{f} or \mathbf{b} at \hat{t} , $DO_i^L(\hat{t}) \cap DO_i^R(\hat{t}) = \emptyset$ for some $i \in \{0, \dots, n_h\}$.

Proof: For $\epsilon > 0$, let $S = \{j : DO_j^L(\hat{t}) \neq DO_j^R(\hat{t})\} \subset \{0, \dots, n_h\}$. If S is a singleton, the proof follows from Theorem 7.3.2. Let us now assume that S is not a singleton. Assume we can find $\mathbf{p}_j^* \in DO_j^L(\hat{t}), \mathbf{p}_j^* \in DO_j^R(\hat{t})$ for all $j \in S$. Following the same analysis as in Theorem 7.3.2, some arguments of absolute value functions would go from either positive or negative values to zero, but no argument to an absolute value function would cross zero. This situation contradicts Corollary 3.15 in [73]. Therefore, there exists at least one j such that $DO_j^L(\hat{t}) \cap DO_j^R(\hat{t}) = \emptyset$. \square

Corollary 7.3.2. Consider ODE (7.1) with $n_h = 0$ (LP instead of LLP) and let Assumption 7.3.1 hold. Let $t_j, t_k \in [t_0, t_f]$ and $[t_j, t_k] \cap Z_t = \hat{t}$, where Z_t is the finite set that contains points at which $DO^L(t) \neq DO^R(t)$ (see Corollary A.5 in [73]). For appropriate $\epsilon > 0$, let $DO(t) = DO^L(\hat{t})$ for all $t \in (\hat{t} - \epsilon, \hat{t}) \equiv T_L$ and $DO(t) = DO^R(\hat{t})$ for all $t \in (\hat{t}, \hat{t} + \epsilon) \equiv T_R$. Let $S(t)$ be the set containing all optimal bases at time t . If no argument to an absolute value function crosses zero in an abs-factorable representation of \mathbf{f} or \mathbf{b} at \hat{t} , $S(t_1) \cap S(t_2) = \emptyset$ for any $t_1 \in T_L$ and $t_2 \in T_R$.

Proof: Each basis B corresponds to a primal extreme point \mathbf{v}^* and a dual extreme point \mathbf{p}^* . In the case of an LP parameterized by its right-hand side, the dual extreme point \mathbf{p}^* is constant whereas \mathbf{v}^* varies with variations in the right-hand side. If basis B is optimal, these points are primal and dual feasible, respectively. Let us assume there is basis $B^* \in S(t_1)$ and $B^* \in S(t_2)$. This would mean there is $\mathbf{p}^* \in DO^L(\hat{t})$ and

$\mathbf{p}^* \in DO^R(\hat{t})$. This contradicts Corollary 7.3.2. \square

Corollary 7.3.3. Consider ODE (7.1) with $n_h > 0$ (embedded LLP) and let Assumption 7.3.1 hold. Let $t_j, t_k \in [t_0, t_f]$ and $[t_j, t_k] \cap Z_t = \hat{t}$, where Z_t is the finite set that contains points at which $DO_i^L(t) \neq DO_i^R(t)$ for any $i \in \{0, \dots, n_h\}$ (see Corollary A.5 in [73]). For appropriate $\epsilon > 0$ and for all $i \in \{0, \dots, n_h\}$, let $DO_i(t) = DO_i^L(\hat{t})$ for all $t \in (\hat{t} - \epsilon, \hat{t}) \equiv T_L$ and $DO_i(t) = DO_i^R(\hat{t})$ for all $t \in (\hat{t}, \hat{t} + \epsilon) \equiv T_R$. Let $S_i(t)$ be the set containing all optimal bases at time t for the i th level of the embedded LLP. If no argument to an absolute value function crosses zero in an abs-factorable representation of \mathbf{f} or \mathbf{b} at \hat{t} , $S_i(t_1) \cap S_i(t_2) = \emptyset$ for any $t_1 \in T_L$ and $t_2 \in T_R$ for some $i \in \{0, \dots, n_h\}$.

Proof: Let $K = \{j : DO_j^L(\hat{t}) \neq DO_j^R(\hat{t})\} \subset \{0, \dots, n_h\}$. If K is a singleton, the proof follows from Corollary 7.3.2. If K is not a singleton, assume we can find bases $B_j^* \in S_j(t_1)$ and $B_j^* \in S_j(t_2)$ for all $j \in K$. This would mean there is $\mathbf{p}_j^* \in DO_j^L(\hat{t})$ and $\mathbf{p}_j^* \in DO_j^R(\hat{t})$ for all $j \in K$. This contradicts Corollary 7.3.1. \square

Theorem 7.3.3. Consider ODE (7.1) and let Assumption 7.3.1 hold. Let $t_j, t_k \in [t_0, t_f]$ and $[t_j, t_k] \cap Z_t = \hat{t}$, where Z_t is the finite set that contains points at which $DO_i^L(t) \neq DO_i^R(t)$ for any $i \in \{0, \dots, n_h\}$ (see Corollary A.5 in [73]). For appropriate $\epsilon > 0$ and for all $i \in \{0, \dots, n_h\}$, let $DO_i(t) = DO_i^L(\hat{t})$ for all $t \in (\hat{t} - \epsilon, \hat{t}) \equiv T_L$ and $DO_i(t) = DO_i^R(\hat{t})$ for all $t \in (\hat{t}, \hat{t} + \epsilon) \equiv T_R$. Let $S(t)$ be the set of possible optimal bases of the LLP returned by Algorithm 2 in [56] at time t (each element of $S(t)$ represents a single basis for all levels of the LLP). Assume no argument to an absolute value function in an abs-factorable representation of \mathbf{f} or \mathbf{b} crosses zero at \hat{t} . Then, $S(t_1) \cap S(t_2) = \emptyset$ for any $t_1 \in T_L$ and $t_2 \in T_R$.

Proof: If $n_h = 0$, the Theorem follows directly from Corollary 7.3.2. When $n_h > 0$, Algorithm 2 in [56] finds a single basis matrix $B^* \in S(t)$ that describes the solution of LPs (7.2) and (7.3) at time t . Let $S_i(t)$ be the set containing all optimal bases at time t for the i th level of the embedded LLP. This single basis can be related to optimal bases $B_i^* \in S_i(t)$ for each level i of the LLP. A basis B_i^* becomes infeasible when $\mathbf{v}_{B_i^*} \not\geq \mathbf{0}$. Theorem 2 in [56] shows that for each i , the solution vector has the

following structure:

$$\mathbf{v}_{B_i^*} = \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_{B^*}^{-1} \mathbf{b}(\mathbf{x}(t, \mathbf{p})) \end{bmatrix}, \quad (7.30)$$

with $\mathbf{0} \in \mathbb{R}^i$. Therefore, for all $i > 0$, any basis B_i^* becoming primal infeasible corresponds to B^* becoming primal infeasible.

Corollary 7.3.3 indicates that there exists at least one level i such that $S_i(t_1) \cap S_i(t_2) = \emptyset$ for $t_1 \in T_L, t_2 \in T_R$. Then, B_i^* becomes infeasible for some i after time \hat{t} . Then B^* also becomes infeasible because of the structure of (7.30). \square

Remark 7.3.2. Another way of proving Theorem 7.3.3 is by noticing that all bases B_i^* are describing the same solution point, and since optimality conditions are not changing with time, this solution point will be optimal as long as it is feasible, where the feasibility conditions are equal for all bases B_i^* .

Corollary 7.3.4. Consider ODE (7.1) and let Assumption 7.3.1 hold. Let $Z_t \subset [t_0, t_f]$ be the finite set that contains points at which $DO_i^L(t) \neq DO_i^R(t)$ for any $i \in \{0, \dots, n_h\}$ (see Corollary A.5 in [73]). Then all times in Z_t correspond to at least one of the following:

1. An argument of an absolute value function in an abs-factorable representation of \mathbf{f} or \mathbf{b} crossing zero.
2. The optimal LLP basis in the sense of Algorithm 2 in [56] becoming infeasible.

Proof: When no arguments of absolute value functions in an abs-factorable representation of \mathbf{f} or \mathbf{b} cross zero, the proof follows directly from Theorem 7.3.3.

Assume that an LLP basis in the sense of Algorithm 2 in [56] remains optimal on $T \equiv (t_j, t_k) \subset [t_0, t_f]$. There could still exist $\hat{t} \in T$ such that $DO_i^L(\hat{t}) \neq DO_i^R(\hat{t})$ for some $i \in \{0, \dots, n_h\}$. In particular, let us for some i assume that $DO_i^L(\hat{t}) \neq DO_i^R(\hat{t})$ but an LLP basis remains optimal. This would mean there exists an optimal dual extreme point \mathbf{p}_i^* for some i that belongs to $DO_i^L(\hat{t})$ and $DO_i^R(\hat{t})$. Select an appropriate $\epsilon > 0$ such that $DO_i(t_1) = DO_i^L(\hat{t})$ and $DO_i(t_2) = DO_i^R(\hat{t})$ for all

$t_1 \in (\widehat{t} - \epsilon, \widehat{t}) \equiv T_L$ and all $t_2 \in (\widehat{t}, \widehat{t} + \epsilon) \equiv T_R$, respectively. Taking the abs-factorable representation of the embedded LPs, at least one of the following two situations is taking place:

1. $\mathbf{k}^* \in DO_i^L(\widehat{t})$, $\mathbf{k}^* \notin DO_i^R(\widehat{t})$. Then, $(\mathbf{p}^* - \mathbf{k}^*)^\top \mathbf{q}^i(\mathbf{z}_{t_1}) = 0$, $(\mathbf{p}^* - \mathbf{k}^*)^\top \mathbf{q}^i(\mathbf{z}_{t_2}) > 0$.
2. $\mathbf{k}^* \notin DO_i^L(\widehat{t})$, $\mathbf{k}^* \in DO_i^R(\widehat{t})$. Then, $(\mathbf{p}^* - \mathbf{k}^*)^\top \mathbf{q}^i(\mathbf{z}_{t_1}) > 0$, $(\mathbf{p}^* - \mathbf{k}^*)^\top \mathbf{q}^i(\mathbf{z}_{t_2}) = 0$.

For some level i , we would have absolute value functions going from positive or negative values to zero value at exactly \widehat{t} . From Corollary 3.15 in [73] these changes will require an argument to an absolute value function to cross zero at \widehat{t} . This can only happen in an abs-factorable representations of \mathbf{f} or \mathbf{b} . \square

Remark 7.3.3. Consider ODE (7.1) with n embedded LLPs, $n > 1$ and let $B^i(t)$ be an optimal basis of the i th LLP returned by Algorithm 2 in [56] at time t . Let Assumption 7.3.1 hold for each one of the embedded LLPs. Then the times in Z_t correspond to $B^i(t)$ becoming infeasible for some $i \in \{1, \dots, n\}$ or an argument to an absolute value crossing zero in the abs-factorable representation of \mathbf{f} or \mathbf{b} .

Corollary 7.3.5. Consider ODE 7.1 and let Assumption 7.2.1 hold. Let $t_j, t_k \in Z_t$ with $t_j < t_k$ and $(t_j, t_k) \cap Z_t = \emptyset$. Let $t^* \in (t_j, t_k)$ such that $DO_i^R(t^*) = DO_i(t^*) = DO_i^L(t^*)$ for all $i = 0, \dots, n_h$. Let $\mathbf{z}_t = \mathbf{x}(t, \mathbf{p})$ and let $\mathbf{z}^* = \mathbf{x}(t^*, \mathbf{p})$. Then for almost every $t \in (t_j, t_k)$ and for all $i = 0, \dots, n_h$,

$$\begin{aligned} (\mathbf{A}^i)^\top \boldsymbol{\lambda} &\leq \mathbf{c}_i, \\ -(\mathbf{q}^i(\mathbf{z}_t))^\top \boldsymbol{\lambda} &\leq -h_i(\mathbf{z}_t), \end{aligned} \tag{7.31}$$

and

$$\begin{aligned} (\mathbf{A}^i)^\top \boldsymbol{\lambda} &\leq \mathbf{c}_i, \\ -(\mathbf{q}^i(\mathbf{z}^*))^\top \boldsymbol{\lambda} &\leq -h_i(\mathbf{z}^*), \end{aligned} \tag{7.32}$$

contain the same extreme points.

Proof: For all i , (7.31) is a representation of $Y_i^*(t)$ (the dual solution set of LPs (7.2)

or (7.3) at time t). The extreme points of $Y_i^*(t)$ correspond to $DO_i(t)$. Since $DO_i(t)$ is constant almost everywhere in $t \in (t_j, t_k)$, the equivalence follows. \square

Corollary 7.3.6. Consider ODE 7.1 and let Assumption 7.2.1 hold. Consider the LLP (7.11) which gives the directional derivative of a LLP parameterized by some components of its right-hand side. Let $t_l, t_k \in Z_t$ with $t_l < t_k$ and $(t_l, t_k) \cap Z_t = \emptyset$. Let $t^* \in (t_l, t_k)$ such that $DO_i^R(t^*) = DO_i(t^*) = DO_i^L(t^*)$ for all $i = 0, \dots, n_h$. Let $\mathbf{z}_t = \mathbf{x}(t, \mathbf{p})$ and assume $\mathbf{z}_t \in \text{int}(F)$ and $\widehat{\mathbf{b}}(\mathbf{z}_t) \in \text{int}(\tilde{F})$ for all t . Let $\mathbf{z}^* = \mathbf{x}(t^*, \mathbf{p})$. Then for almost all $t \in (t_l, t_k)$ and for $0 \leq i \leq n_h$ both

$$\begin{aligned} [h_i]'(\mathbf{z}_t; \mathbf{d}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v+1}} \begin{bmatrix} \mathbf{c}_i^T & -h_i(\mathbf{z}_t) \end{bmatrix} \mathbf{v}, \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{A}^i & -\mathbf{q}^i(\mathbf{z}_t) \end{bmatrix} \mathbf{v} = [\mathbf{q}^i]'(\mathbf{z}_t; \mathbf{d}), \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{7.33}$$

and

$$\begin{aligned} \min_{\mathbf{v} \in \mathbb{R}^{n_v+1}} & \begin{bmatrix} \mathbf{c}_i^T & -h_i(\mathbf{z}^*) \end{bmatrix} \mathbf{v}, \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{A}^i & -\mathbf{q}^i(\mathbf{z}^*) \end{bmatrix} \mathbf{v} = [\mathbf{q}^i]'(\mathbf{z}_t; \mathbf{d}), \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{7.34}$$

attain the same objective function value (note that \mathbf{z}_t appears in the right-hand side of (7.34)), if (7.34) is feasible. Therefore for almost all $t \in (t_l, t_k)$, the objective function values of LLP (7.11) for $j = 0$ can be computed by solving LLP (7.34), which is an LLP parameterized by some components of its right-hand side.

Proof: From [47], $\tilde{\mathbf{h}}$ is directionally differentiable and l -smooth on $\text{int}(\tilde{F})$ and since $\widehat{\mathbf{b}}$ is abs-factorable, \mathbf{h} is l -smooth at $\mathbf{z} \in \text{int}(F)$ if $\widehat{\mathbf{b}}(\mathbf{z}) \in \text{int}(\tilde{F})$. Since $\mathbf{z}_t \in \text{int}(F)$ and $\widehat{\mathbf{b}}(\mathbf{z}_t) \in \text{int}(\tilde{F})$ for all t , h_i is directionally differentiable and l -smooth at \mathbf{z}_t for all i and (7.33) always attains a finite solution. Since (7.33) always attain a finite solution, (7.34) is always dual feasible. Therefore, (7.34) must either attain a finite solution or be infeasible.

From strong duality, the duals of LPs (7.33) and (7.34) attain the same objective function values, respectively. The feasible sets of the duals of (7.33) and (7.34) are described in Corollary 7.3.5, and this same Corollary argues that these sets share the same extreme points almost everywhere. By Theorem 2.7 in [12], the solution sets of the duals of (7.33) and (7.34) will include an extreme point. Therefore, the objective function values of both LPs can be computed as $\max_{\lambda \in DO_i} \boldsymbol{\lambda}^T [\mathbf{q}^i]'(\mathbf{z}_t; \mathbf{d})$. Then, the same objective function value is attained by LPs (7.33) and (7.34) almost everywhere in (t_l, t_k) . \square

The fact that LP (7.34) can become infeasible is not a problem. Consider B to be an optimal basis of LP (7.34) at time t^* . If LP (7.34) becomes infeasible, then basis B must become infeasible too, which can be detected reliably using the strategies in [56]. Every time an optimal basis of (7.34) becomes infeasible, the LP (7.34) can be reconstructed using (7.33) which will always be well-defined.

Remark 7.3.4. LP (7.34) is an LP parameterized by its right-hand side.

The sensitivity system for ODE (7.1) is given by the ODE system (7.6). Given that LLP (7.11) with $j = 0$ can be reformulated as a finite time sequence of LLPs parameterized by some components of their right-hand sides, the ODE system giving the directional derivative of ODE (7.1) can be reformulated as a finite time sequence of ODE systems with LLPs parameterized by some components of their right-hand sides. Since LD-derivatives are obtained by taking the directional derivative of the previous column, the same analysis applies for the rest of the LD-derivatives. This is expressed in the following two Remarks.

Remark 7.3.5. Consider ODE (7.1) and let $\mathbf{x}_t(\mathbf{p}) \equiv \mathbf{x}(t, \mathbf{p})$. Let $\mathbf{x}_t(\mathbf{p}) \in \text{int}(F)$ and $\widehat{\mathbf{b}}(\mathbf{x}_t(\mathbf{p})) \in \text{int}(\tilde{F})$ for all $t \in [t_0, t_f]$. Then $\mathbf{x}'_t(\mathbf{p}; \mathbf{d})$ is given by the solution of a finite time sequence of ODE systems with embedded LLPs parameterized by their right-hand sides and can be reformulated as a finite time sequence of DAE systems using the strategies in [56]. The sets containing the extreme points of the dual feasible sets of the embedded LLPs can change at each time $t \in Z_t$. Therefore, a new LLP must be constructed and a new optimal basis computed at every $t \in Z_t$.

Remark 7.3.6. Consider ODE (7.1) and let $\mathbf{x}_t(\mathbf{p}) \equiv \mathbf{x}(t, \mathbf{p})$. Let $\mathbf{x}_t(\mathbf{p}) \in \text{int}(F)$ and $\widehat{\mathbf{b}}(\mathbf{x}_t(\mathbf{p})) \in \text{int}(\widetilde{F})$ for all $t \in [t_0, t_f]$. Let $\mathbf{M} \in \mathbb{R}^{n_x \times q}$ and consider $[\mathbf{x}_t]_{\mathbf{p}_0, \mathbf{M}}^{(k-1)}(\mathbf{d})$ with $k \in \{1, \dots, q\}$ which is given by an ODE system with a LLP embedded parameterized by its right-hand side and technology matrix. Then, there exists a finite set $Z_t^k \subset [t_0, t_f]$ such that for $t_i, t_j \in Z_t^k$, $(t_i, t_j) \cap Z_t^k = \emptyset$, the embedded LLP in the ODE sensitivity system (7.6) that gives $[\mathbf{x}_t]_{\mathbf{p}_0, \mathbf{M}}^{(k-1)}(\mathbf{d})$ can be reformulated as a LLP parameterized only by its right-hand side almost everywhere in (t_i, t_j) .

The fact that LP (7.34) is equivalent to LP (7.33) almost everywhere is not a problem either. The points where they are different are of measure zero. Therefore, these points have no impact on the solution of an ODE system. However, any good methods will rely on detecting when LP (7.34) is constructed at any of these points. In the unfortunate situation LP (7.34) is constructed at one of these points, the computation of sensitivities can be wrong. Example 7.3.2 shows what could happen in a very similar situation. In Example 7.3.2, LP (7.34) is constructed at $t \in Z_t$ leading to the wrong sensitivities. Care must be taken if integration is started or restarted at t^* and for any i , $DO_i(t^*) \neq DO_i^R(t^*)$ for all small enough $\epsilon > 0$.

The observation in Remark 7.3.1 can be useful in this case. If LP (7.34) is constructed at $\widehat{t} \in Z_t$ (or any \widehat{t} such that $DO_i(\widehat{t}) \neq DO_i^L(\widehat{t})$ when $DO_i^L(\widehat{t}) = DO_i^R(\widehat{t})$ for some $i \in \{0, \dots, n_h\}$), the basis B^* obtained using Algorithm 2 in [56] could be describing $\boldsymbol{\lambda}_i^* \in DO_i(\widehat{t})$, $\boldsymbol{\lambda}_i^* \notin DO_i^R(\widehat{t})$. In this case, a violation of the dual constraints $-(\mathbf{q}^i(\mathbf{z}_t))^T \boldsymbol{\lambda} \leq -h_i(\mathbf{z}_t)$ in (7.31) would happen at time $\widehat{t} + \epsilon$ for all small enough $\epsilon > 0$ and some level $i \in \{0, \dots, n_h\}$. This specific dual constraints are also given by the last component of the reduced cost vectors for each level of LLP (7.33). When solving the embedded LLP at time t^* , the optimal dual vertices $\boldsymbol{\lambda}_i^*$, which are time-invariant, can be extracted for each level $i \in \{0, \dots, n_h\}$. Then, the time-derivatives of the reduced costs can be inspected (if the time-derivatives are continuous in time) or the last component of the reduced costs vector for each level $i \in \{0, \dots, n_h\}$ can be integrated for a very short period of time and verify that they remain positive. This problem is illustrated in Example 7.3.2.

Definition 7.3.3. Consider ODE (7.1) and let Assumption 7.3.1 hold for all embedded LLPs. In addition, let $\mathbf{x}(t, \mathbf{p}) \in \text{int}(F)$ and $\widehat{\mathbf{b}}(\mathbf{x}(t, \mathbf{p})) \in \text{int}(\tilde{F})$ for all $t \in [t_0, t_f]$ for all embedded LLPs. Let Z_t^0 include all points at which an LLP basis becomes infeasible in any embedded LLP and any *valley 0-crossing* as in [71] for any absolute value function in the abs-factorable representation of \mathbf{f} or \mathbf{b} . For $k \in \{1, \dots, q\}$, let Z_t^k contain all elements in Z_t^{k-1} , any times at which an LLP basis becomes infeasible as in [56] in the k^{th} column of the sensitivity ODE system in (7.6), and any *valley k-crossing* as in [71] for any absolute value function in the abs-factorable representation of \mathbf{f} or \mathbf{b} .

Therefore, the embedded LLPs in ODE (7.6) can all be transformed into LLPs parameterized by their right-hand sides and these ODEs can be reformulated as a finite time sequence of DAEs using the strategies in [56]. In particular for $\mathbf{M} \in \mathbb{R}^{n_x \times q}$, $[\mathbf{x}_t]_{\mathbf{p}, \mathbf{M}}^{(k)}$ will be solutions to ODE systems with embedded LLPs parameterized by their technology matrix and right-hand side for each $k \in \{0, \dots, q-1\}$. These ODE systems can be reformulated as a finite time sequence of ODE systems with embedded LLPs parameterized by their right-hand sides. For each $k \in \{0, \dots, q-1\}$, Z_t^k contains the times at which the embedded LLPs parameterized by their right-hand sides needs to be reconstructed when integrating the ODE system associated with $[\mathbf{x}_t]_{\mathbf{p}, \mathbf{M}}^{(k)}$.

Example 7.3.2. Consider Example 7.3.1, let $\mathbf{p}_0 = (1, 0)$ and $\mathbf{d} = (0, -1)$. Then

$$\mathbf{y}(t, \mathbf{p}_0) = (1, \ln(t + e) - 1). \quad (7.35)$$

The directional derivative $\mathbf{y}'_t(\mathbf{p}; \mathbf{d})$ is given by the solution to ODE (7.25). The use of Corollary 7.3.6 can lead to very different results to the sensitivities. This is because $DO(0) \neq DO(\epsilon)$ for all small enough $\epsilon > 0$. From (7.35) and LP (7.23) it can be observed that $DO(0) = \{(1, 1), (1, 0)\}$ whereas for all small enough $\epsilon > 0$, $DO(\epsilon) = \{(1, 1)\}$.

The solution to

$$h'(\mathbf{y}(0, \mathbf{p}_0); \mathbf{y}'_0(0, \mathbf{p}_0; 0, \mathbf{d})) = \max_{\boldsymbol{\lambda} \in DO(0)} [0 \quad -1] \boldsymbol{\lambda} = 0,$$

with an optimal vertex $\boldsymbol{\lambda}^* = (1, 0)$. However, $\boldsymbol{\lambda}^* \notin DO(\epsilon)$. If Corollary 7.3.6 is used without checking whether $DO(0) = DO(\epsilon)$ for all small enough $\epsilon > 0$, no events are detected and the right plot of Figure 7-1 is obtained.

The description of $DO(t)$ is the following:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ -y_1(t, \mathbf{p}_0) & -y_2(t, \mathbf{p}_0) \end{bmatrix} \boldsymbol{\lambda} \leq \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ -h(\mathbf{y}(t, \mathbf{p}_0)) \end{bmatrix},$$

where the last row is always satisfied as an equality (it is an implicit equality constraint). For small enough $\epsilon > 0$, $\boldsymbol{\lambda}^* \in DO(\epsilon)$ if

$$-\left(\frac{d\mathbf{y}}{dt}(0, \mathbf{p}_0)\right)^T \boldsymbol{\lambda}^* = -\left(\frac{dy_1}{dt}(0, \mathbf{p}_0)\right) < -h'(\mathbf{y}(0, \mathbf{p}_0); \frac{d\mathbf{y}(0, \mathbf{p}_0)}{dt})$$

is satisfied and $\boldsymbol{\lambda}^* \notin DO(\epsilon)$ if

$$-\left(\frac{d\mathbf{y}}{dt}(0, \mathbf{p}_0)\right)^T \boldsymbol{\lambda}^* = -\left(\frac{dy_1}{dt}(0, \mathbf{p}_0)\right) > -h'(\mathbf{y}(0, \mathbf{p}_0); \frac{d\mathbf{y}(0, \mathbf{p}_0)}{dt}).$$

A definitive conclusion cannot be drawn if they are equal and integration for a small period of time will be required. Now,

$$\frac{d\mathbf{y}}{dt}(0, \mathbf{p}_0) = [0 \quad e^{-h(\mathbf{y}(0, \mathbf{p}_0))}] = [0 \quad e^{-1}],$$

and then

$$h'(\mathbf{y}(0, \mathbf{p}_0); [0 \ e^{-1}]) = \max_{\boldsymbol{\lambda} \in DO(0)} [0 \ e^{-1}] \boldsymbol{\lambda} = e^{-1}.$$

Since $-\left(\frac{d\mathbf{y}}{dt}(0, \mathbf{p}_0)\right)^T \boldsymbol{\lambda}^* = 0 > h'(\mathbf{y}(0, \mathbf{p}_0); [0 \ e^{-1}]) = -e^{-1}$, $\boldsymbol{\lambda}^* \notin DO(\epsilon)$. Therefore, LP (7.34) should be constructed using $DO(\epsilon)$ instead of $DO(0)$ for small enough $\epsilon > 0$. A way to deal with this specific problem is to verify dual feasibility using event detection. To do this, verify that $h(\mathbf{y}(t^*, \mathbf{p}_0)) - \mathbf{y}(t^*, \mathbf{p}_0)^T \boldsymbol{\lambda}^* \leq 0$.

The left plot in Figure 7-1 shows the correct sensitivities obtained solving LPs (7.22) and (7.33) at each time point. This problem was solved using Gurobi [50] with optimality and feasibility tolerances of 10^{-8} for the LP. The absolute and relative integration tolerances used were of 10^{-7} . If instead $\mathbf{p}_0 = (1, -10^{-6})$, all three methods produce the same plot because now $DO(0) = DO(\epsilon)$ for all small enough $\epsilon > 0$.

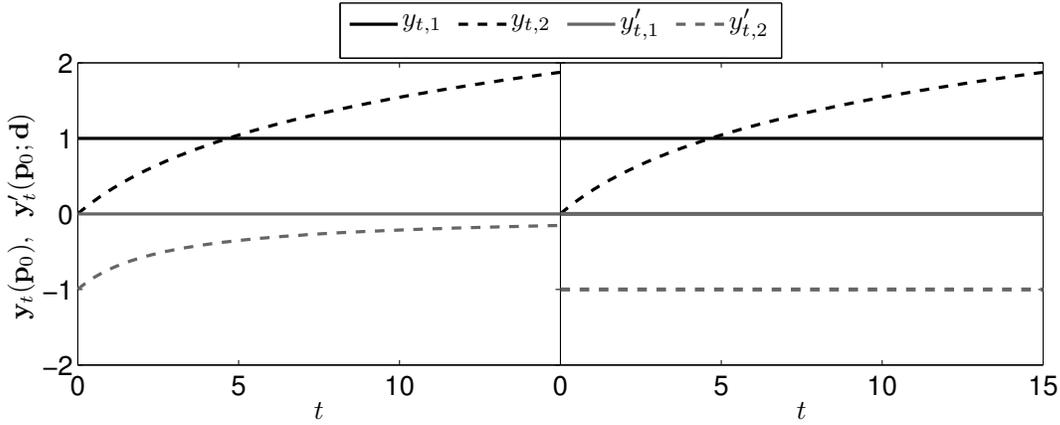


Figure 7-1: Sensitivities plots for Example 7.3.2. Left: Correct sensitivities plot for $\mathbf{p}_0 = (1, 0)$, $\mathbf{d} = (0, -1)$. Right: Plot obtained if $DO(0)$ is used as the feasible set of the directional derivative LP. This example shows that although $DO(t)$ is constant for any $t > 0$, it is different at $t = 0$.

In the case of $\mathbf{p}_0 = [1, 0]$ and $T = [0, 15]$, solving for \mathbf{y} and \mathbf{y}' by solving LPs (7.22) and (7.33) at each time step takes about 0.33 seconds, whereas using the algebraic reformulation for both \mathbf{y} and \mathbf{y}' results in approximately 0.039 seconds, a gain of

almost an order of magnitude.

Now, let us revisit the case when $\mathbf{p}_0 = [1, -1]$ which encounters a point of singularity in the directional derivative LP. Using Corollary 7.3.6 we can find that the optimal basis at $t = 0$ for LP (7.24) is

$$\mathbf{B}(0) = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{c}_B(0) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad (7.36)$$

and $h'(\mathbf{y}(0, \mathbf{p}_0); \mathbf{s}(0)) = 0$. Since $\mathbf{s}(t) = (0, 1)$ for all $t \in (0, e)$, $h'(\mathbf{y}(t, \mathbf{p}_0); \mathbf{s}(t)) = 0$ for all $t \in (0, e)$. Then, using LP (7.24) with $\mathbf{z} = \mathbf{y}(t, \mathbf{p}_0)$ or with $\mathbf{z} = \mathbf{y}(0, \mathbf{p}_0)$ for $t \in (0, e)$ yields the same result. The advantage of using LP (7.24) with $\mathbf{z} = \mathbf{y}(0, \mathbf{p}_0)$ is that the optimal basis (7.36) does not become ill-conditioned or singular. Nevertheless at $t = e$, LP (7.24) evaluated at $\mathbf{z} = \mathbf{y}(0, \mathbf{p}_0)$ ceases to be valid. This is detected by a change of basis at $t = e$ for LP (7.22). In particular, $B(t) = \{\{1, 4\}\}$ for $t \in (0, e)$, $B(t) = \{\{1, 2\}\}$ for $t \in (e, 15)$, and $B(e) = \{\{1, 2\}, \{1, 4\}\}$ where $B(t)$ is the set of optimal bases at time t .

Algorithm 2 computes the LD-derivatives of LLPs parameterized by some components of their right-hand side in a more efficient way. It relies on detecting whether a given basis is compatible with the set of directions \mathbf{M} . This “compatibility” may be considered an extension of the theory in [48]. When a basis is compatible with a set of directions \mathbf{M} , then it can be used to calculate the LD-derivatives without solving LLP (7.11).

The theory in [71] for efficient integration of the ODE sensitivity system relies on being able to detect all times Z_t^k for all k as in Definition 7.3.3. In particular, the times in Z_t^k correspond to times where arguments to absolute value functions cross zero; at this times, some arguments of absolute value functions can enter sliding modes [71]. Being able to detect sliding modes is important for efficient numerical integration. If LLP (7.11) is not solved at time t^* because an optimal basis B of LLP (7.4) is compatible with the directions $\widehat{\mathbf{b}}'(\mathbf{x}_{t^*}(\mathbf{p}); \mathbf{x}'_{t^*}(\mathbf{p}; \mathbf{M}))$, a basis change of LLP (7.11) can still be detected, which in turn means a time in Z_t^k can be detected. This

is shown in Theorem 7.3.4.

Theorem 7.3.4. Consider ODE (7.1) in $T = [t_0, t_f]$ and let $\mathbf{x}(t, \mathbf{p}) \in \text{int}(F)$ and $\widehat{\mathbf{b}}(\mathbf{x}(t, \mathbf{p})) \in \text{int}(\widetilde{F})$ for all $t \in T$. Let $\mathbf{x}_t(\mathbf{p}) \equiv \mathbf{x}(t, \mathbf{p})$ and consider $\mathbf{x}'_t(\mathbf{p}; \mathbf{d})$, which can be computed as the solution of a time sequence of ODE systems with LLPs embedded parameterized by some elements of their right-hand sides. Let $B(t)$ be an optimal basis of LLP (7.4) and $B^{(0)}(t)$ an optimal basis of LLP (7.11) returned by Algorithm 2 in [56] at time t . Then, each element of Z_t^1 can be detected as at least one of the following:

1. An element of Z_t ,
2. A *valley-1-crossing* as in [71],
3. A basis change in $B^{(0)}(t)$ as in [56],
4. Basis $B(t)$ becoming not compatible with direction $\widehat{\mathbf{b}}'(\mathbf{x}_t(\mathbf{p}); \mathbf{x}'_t(\mathbf{p}; \mathbf{d}))$.

Proof: The first three situations follow from Definition 7.3.3. This Theorem is relevant when an optimal basis $B(t)$ is found to be compatible with the first direction, and therefore $B^{(0)}(t)$ is not computed. In this case, elements of Z_t^1 corresponding to basis $B^{(0)}(t)$ becoming primal infeasible as in [56] can still be detected. For this, let us assume that for $t_1 \in T$, $Z_{t_1} \cap (t_0, t_1) = \emptyset$, $\widehat{t} \in (t_0, t_1)$, there are no *valley-1-crossings* at time \widehat{t} , basis $B^{(0)}(t_0)$ becomes primal infeasible as in [56] at \widehat{t} , but this basis is not computed because $B(t_0)$ is compatible with direction $\widehat{\mathbf{b}}'(\mathbf{x}_{t_0}(\mathbf{p}); \mathbf{x}'_{t_0}(\mathbf{p}; \mathbf{d}))$.

Let $\widehat{\mathbf{d}}(t, \mathbf{p}) \equiv \widehat{\mathbf{b}}'(\mathbf{x}_t(\mathbf{p}); \mathbf{x}'_t(\mathbf{p}; \mathbf{d}))$. If basis $B(t_0)$ is compatible with direction $\widehat{\mathbf{d}}(t, \mathbf{p})$, the objective function values of LLP (7.11) at time t can be computed as $(\mathbf{C}_{B(t_0)})^T \mathbf{A}_{B(t_0)}^{-1} \widehat{\mathbf{d}}(t, \mathbf{p})$. Since \mathbf{A}^{n_h} is full row rank, the cost vectors are all linearly independent with respect to their technology matrix \mathbf{A}^i , and at each level i , there is at least one strictly positive reduced cost (if all reduced costs are equal to zero, the cost vector is linearly dependent). Choose the index of any such variable for each level i and name it $\alpha_1, \dots, \alpha_{n_h}$, respectively, where they are all different among themselves. Notice that reduced costs only change with a basis change. Then, set

$B^i(t_0) = B(t_0) \cup \alpha_1 \cup \dots \cup \alpha_i$. By construction, bases $B^i(t_0)$ are optimal for each level i , respectively.

From the structure of the primal solution vector in Theorem 2 in [56] (which is described too in Theorem 7.3.3), for $t \in (t_0, \hat{t})$, for all i , and for all $k < i$,

$$\begin{aligned} (\mathbf{C}_{B^i(t_0)})^T \mathbf{A}_{B^i(t_0)}^{-1} [\mathbf{q}^i]'(\mathbf{x}_{t_0}(\mathbf{p}); \mathbf{x}'_{t_0}(\mathbf{p}; \mathbf{d})) &= (\mathbf{C}_{B^k(t_0)})^T \mathbf{A}_{B^k(t_0)}^{-1} [\mathbf{q}^k]'(\mathbf{x}_{t_0}(\mathbf{p}); \mathbf{x}'_{t_0}(\mathbf{p}; \mathbf{d})) \\ &= (\mathbf{C}_{B(t_0)})^T \mathbf{A}_{B(t_0)}^{-1} \hat{\mathbf{d}}(t_0, \mathbf{p}). \end{aligned}$$

For all $i \in \{0, \dots, n_h\}$, let $\boldsymbol{\lambda}^i = ([\mathbf{A}^i]_{B^i(t_0)}^{-1})^T \mathbf{c}_{i, B^i(t_0)}$. For all i , $\boldsymbol{\lambda}^i$ is optimal for LP (7.26) because the basis $B^i(t_0)$ is optimal. Therefore for $t \in (t_0, \hat{t})$, $\boldsymbol{\lambda}^i$ is feasible in LP (7.9) for $j = 0$, and it is optimal because $[\mathbf{q}^i]'(\mathbf{x}_t(\mathbf{p}); \mathbf{x}'_t(\mathbf{p}; \mathbf{d}))^T \boldsymbol{\lambda}^i = \mathbf{c}_{i, B^i(t_0)}^T \mathbf{A}_{B^i(t_0)}^{-1} [\mathbf{q}^i]'(\mathbf{x}_t(\mathbf{p}); \mathbf{x}'_t(\mathbf{p}; \mathbf{d})) = (\mathbf{c}_{i, B(t_0)})^T \mathbf{A}_{B(t_0)}^{-1} \hat{\mathbf{d}}(t, \mathbf{p})$, which corresponds to the objective function value of LP (7.9) because $B(t_0)$ is compatible with direction $\hat{\mathbf{d}}(t, \mathbf{p})$.

Assume that for all $t \in (\hat{t}, t_1)$, $B(t_0)$ is compatible with directions $\hat{\mathbf{d}}(t, \mathbf{p})$, but basis $B^{(0)}(t_0)$ becomes primal infeasible in LLP (7.11). Since by assumption there are no *valley-1-crossings* at \hat{t} as in [71], $DO_i^{(0),L}(\hat{t}) \cap DO_i^{(0),R}(\hat{t}) = \emptyset$ for some i , by Corollary 7.3.1 (where here, $DO_i^{(j)}(t)$ refers to the optimal extreme points of the duals of LLP (7.11) at time t). Then at least for some i , $\boldsymbol{\lambda}_i$ becomes suboptimal, which in turn corresponds with $(\mathbf{c}_{i, B(t_0)})^T \mathbf{A}_{B(t_0)}^{-1} \hat{\mathbf{d}}(t, \mathbf{p})$ not corresponding to $h'_i(\mathbf{x}(t, \mathbf{p}); \hat{\mathbf{d}}(t, \mathbf{p}))$ (the i th objective function value of LLP (7.11)) at time t . Then, $B(t_0)$ can't be compatible with directions $\hat{\mathbf{d}}(t, \mathbf{p})$ for $t \in (\hat{t}, t_1)$ and a contradiction is reached. Then, all basis changes in $B^{(0)}(t)$ must correspond with basis $B(t)$ becoming not compatible with directions $\hat{\mathbf{d}}(t, \mathbf{p})$.

Remark 7.3.7. The same analysis as in Theorem 7.3.4 can be applied for all Z_t^k with $k > 1$.

When using Algorithm 3 to compute the LD-derivatives of \mathbf{h} , the times in Z_t and Z_t^i for all i indicate that the optimal partitions may have changed. The proof is presented in the following Proposition.

Proposition 7.3.5. Consider ODE (7.1) on $[t_0, t_1]$. Let $\mathbf{z}_t(\mathbf{p}) = \mathbf{x}(t, \mathbf{p})$ and let $\mathbf{z}_t(\mathbf{p}) \in \text{int}(F)$ and $\hat{\mathbf{b}}(\mathbf{z}_t(\mathbf{p})) \in \text{int}(\tilde{F})$ for all $t \in [t_0, t_1]$. For $i \in \{0, \dots, n_h\}$, let $\hat{B}^i(t)$

and $\widehat{N}^i(t)$ be the optimal partitions for each level of LLP (7.4) at time t . Then LP (7.20) will attain the same objective function value using $\widehat{B}^i(t)$ and $\widehat{N}^i(t)$ and $\widehat{B}^i(t_0)$ and $\widehat{N}^i(t_0)$ until one of the following happens:

1. The optimal basis of LLP (7.4) becomes infeasible,
2. An argument to an absolute value crosses zero in \mathbf{f} or \mathbf{b} ,
3. For some i , all optimal bases of LP (7.20) becomes infeasible.

Proof: From [47], h_i is directionally differentiable and l -smooth at \mathbf{z}_t for all $t \in [t_0, t_1]$ and for all i and LP (7.20) always attains a finite solution when using $\widehat{B}^i(t)$ and $\widehat{N}^i(t)$. Therefore, when using $\widehat{B}^i(t_0)$ and $\widehat{N}^i(t_0)$, LP (7.20) is always dual feasible, and then it must either attain a finite solution or be primal infeasible.

The dual optimal solution set at time t , $Y_i^*(t)$, can be described by Equation (7.15) using the optimal partition. Notice that $Y_i^*(t)$ is the feasible set of the dual of LP (7.20) and this set is nonempty for all t . If a finite solution is attained, then using $\widehat{B}^i(t_0)$ and $\widehat{N}^i(t_0)$ or $\widehat{B}^i(t)$ and $\widehat{N}^i(t)$ in LP (7.20) must result in the same objective function value because a solution will lie at an extreme point, and $Y_i^*(t)$ and $Y_i^*(t_0)$ contain the same extreme points for all i until either the optimal basis of LLP (7.4) becomes infeasible or an argument to an absolute value crosses zero in \mathbf{f} or \mathbf{b} (Theorem 7.3.3 and Corollary 7.3.4).

LP (7.20) can become dual unbounded when using $\widehat{B}^i(t_0)$ and $\widehat{N}^i(t_0)$, which in turn means it can become primal infeasible. This will result in an optimal basis becoming infeasible, which in turn can be detected reliably using the strategies in [56]. When this happens, a new optimal partition for each level i can be computed. \square

With Proposition 7.3.5, the times at which the optimal partition needs to be updated can be detected reliably. Proposition 7.3.5 refers to directional derivatives only and can be extended to LD-derivatives.

7.4 Integration procedure of ODE systems corresponding to the LD-derivatives of ODE (7.1)

Consider ODE system (7.6) which corresponds to the LD-derivatives of ODE (7.1). As mentioned before, this ODE system does not necessarily satisfy Carathéodory's conditions because $[\mathbf{f}_t]'(\mathbf{x}(t, \mathbf{p}); \cdot)$ can be a discontinuous function [72]. However, an event detection scheme can be used to integrate a time sequence of Carathéodory ODEs.

Let \mathbf{M} be the matrix of directions with q columns. Then, the matrix ODE system (7.6) contains q columns too. Consider ODE (7.1) and the sensitivities ODE (7.6). The superscripts refer to embedded LLPs. Consider that there are n embedded LLPs. Then for $i \in \{1, \dots, n\}$ we have \mathbf{h}^i instead of \mathbf{h} , F^i, \tilde{F}^i instead of F, \tilde{F} , \mathbf{A}^i and n_h^i instead of \mathbf{A} and n_h , and $\mathbf{q}^{i,j}$ instead of \mathbf{q}^j for $j \in \{0, \dots, n_h^i\}$. Algorithm 4 provides a way of computing the sensitivities of ODE systems with n LLPs embedded.

If Algorithm 3 is used, an algorithm analogous to Algorithm 4 can be derived, with the added complication associated with the optimal partition method.

7.5 Numerical Examples

The following examples have been carried out on MATLAB 8.4.0 R2014b on an Intel 2.60 GHz processor using Gurobi 6.0 as the LP solver.

7.5.1 *E. coli* cultivation system

This example is based on Figure 1 in [51], and its sensitivities have been analyzed in Figure 2 in [60]. It consists of a batch reactor where *E. coli* is growing on glucose and xylose. Oxygen concentration is controlled and assumed to be constant. The

Algorithm 4 Method for finding the LD-derivatives of ODE systems with n LLPs embedded.

- 1: Require $\mathbf{x}(t, \mathbf{p}) \in \text{int}(F^i)$ and $\widehat{\mathbf{b}}^i(\mathbf{x}(t, \mathbf{p})) \in \text{int}(\widetilde{F}^i)$ for all $t \in [0, t_f]$ and for all n LLPs.
 - 2: Require $\mathbf{M} \in \mathbb{R}^{n_x \times q}$, $t^* = 0$, $\mathbf{x}(0, \mathbf{p}) = \mathbf{p}$, $\mathbf{S}(0) = \mathbf{M}$.
 - 3: **procedure** CALCULATE
 - 4: **while** $t^* < t_f$ **do**
 - 5: **for** $i = 1 : n$ **do**
 - 6: For LLP i , compute γ^i , the optimal bases $B_0^i, B_1^i, \dots, B_q^i$ for LLP i , the matrices $\mathbf{\Lambda}^{i,j}$, and the vectors $\boldsymbol{\lambda}_{k,l}^i$ for $j \in \{0, \dots, q\}$, $k \in \{0, \dots, n_h^i\}$, and $l \in \{1, \dots, \gamma^i\}$ using Algorithm 2.
 - 7: Compute the technology matrices \mathbf{A}^i , and $\mathbf{A}_{j+1}^i \equiv \mathbf{A}_{\mathbf{x}(t^*, \mathbf{p}), \mathbf{S}(t^*)}^{i,(j)}$ for $j \in \{0, \dots, q-1\}$.
 - 8: Consider $\mathbf{K}^i(t) \equiv \begin{bmatrix} [\mathbf{A}_{B_0^i}^i]_0^{-1}(\mathbf{b}^i(\mathbf{x}(t, \mathbf{p}))) & [\mathbf{A}_{1, B_1^i}^i] \mathbf{s}_1(t) & \dots & [\mathbf{A}_{q, B_q^i}^i] \mathbf{s}_q(t) \end{bmatrix}$.
 - 9: Let $\mathbf{K}_1^i(t) \equiv [\mathbf{k}_1^i(t) \dots \mathbf{k}_\gamma^i(t)]$ and $\mathbf{K}_2^i(t) \equiv [\mathbf{k}_{\gamma+1}^i(t) \dots \mathbf{k}_{q+1}^i(t)]$.
 - 10: **end for**
 - 11: Integrate ODE (7.1) and sensitivities ODE (7.6) using the theory in [71] with $\mathbf{h}^i(\mathbf{x}(t, \mathbf{p})) = \mathbf{\Lambda}^{i,0} \mathbf{b}(\mathbf{x}(t, \mathbf{p}))$ and for $j \in \{0, \dots, q-1\}$, $[\mathbf{h}^i]_{\mathbf{x}(t, \mathbf{p}), \mathbf{S}(t)}^{i,(j)}(\mathbf{s}_{j+1}(t)) = \mathbf{\Lambda}^{i,j+1} \mathbf{s}_{j+1}(t)$ for $i \in \{1, \dots, n\}$ until time \widehat{t} such that:
 - (a) $\mathbf{K}_1^i(\widehat{t}) \not\geq \mathbf{0}$ for some $i \in \{0, \dots, n\}$.
 - (b) $f\text{sign}(\mathbf{K}_2^i(\widehat{t})) \not\geq \mathbf{0}$ for some $i \in \{0, \dots, n\}$ where $f\text{sign}$ applied to a matrix is equal to applying $f\text{sign}$ to each row.
 - (c) For all i such that $\gamma^i > 0$ with $j \in \{0, \dots, n_h^i\}$, $i \in \{1, \dots, n\}$, $k \in \{2, \dots, \gamma^i\}$, $[\boldsymbol{\lambda}_{j,1}^i]^T \mathbf{q}^{i,j}(\mathbf{x}(\widehat{t}, \mathbf{p})) - h_j^i(\mathbf{x}(\widehat{t}, \mathbf{p})) < 0$ and if $\gamma^i > 1$, $[\boldsymbol{\lambda}_{j,k}^i]^T [\mathbf{q}^{i,j}]_{\mathbf{x}(\widehat{t}, \mathbf{p}), \mathbf{S}(\widehat{t})}^{(k-2)}(\mathbf{s}_{k-1}(\widehat{t})) - [\mathbf{h}^i]_{\mathbf{x}(t, \mathbf{p}), \mathbf{S}(\widehat{t})}^{i,(j)}(\mathbf{s}_{j+1}(\widehat{t})) < 0$.
 - (d) An argument to an absolute value function in the abs-factorable representation of \mathbf{f} or \mathbf{b}^i crosses zero.
 - (e) $\widehat{t} = t_f$.
 - 12: $t^* = \widehat{t}$.
 - 13: **end while**
 - 14: **return** $\mathbf{x}(t_f, \mathbf{p}), \mathbf{S}(t_f)$.
 - 15: **end procedure**
-

dynamic equations describing this system are:

$$\dot{y}(t, \mathbf{p}) = \mu(\mathbf{x}(t, \mathbf{p}), \mathbf{p})y(t, \mathbf{p}),$$

$$\dot{g}(t, \mathbf{p}) = -v_g(\mathbf{x}(t, \mathbf{p}), \mathbf{p})y(t, \mathbf{p}),$$

$$\dot{z}(t, \mathbf{p}) = -v_z(\mathbf{x}(t, \mathbf{p}), \mathbf{p})y(t, \mathbf{p}),$$

$$\dot{e}(t, \mathbf{p}) = v_e(\mathbf{x}(t, \mathbf{p}), \mathbf{p})y(t, \mathbf{p}),$$

$$\dot{\alpha}(t, \mathbf{p}) = \gamma(\mathbf{x}(t, \mathbf{p}), \mathbf{p}),$$

where y, g, z, e, α represent the concentrations of biomass, glucose, xylose, ethanol, and the penalty state, respectively, $\mathbf{x}(t, \mathbf{p}) = [y, g, z, \alpha]$, $\mathbf{p} = [y_0, g_0, z_0, e_0, O_0]$, and γ represents the objective function value of the Phase I LP. The quantities $\mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})' = [\gamma(\mathbf{x}(t, \mathbf{p}), \mathbf{p}), \mu(\mathbf{x}(t, \mathbf{p}), \mathbf{p}), v_e(\mathbf{x}(t, \mathbf{p}), \mathbf{p}), v_g(\mathbf{x}(t, \mathbf{p}), \mathbf{p}), v_z(\mathbf{x}(t, \mathbf{p}), \mathbf{p})]$ are given as solutions of the following lexicographic linear program

$$\begin{aligned} \mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) &= \text{lex min } \mathbf{C}^T \mathbf{v}, \\ \text{s.t. } \mathbf{S} \mathbf{v} &= \mathbf{0}, \\ \mathbf{v}^{LB}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) &\leq \mathbf{v} \leq \mathbf{v}^{UB}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}), \end{aligned}$$

where \mathbf{C} contains the following objectives:

1. minimize Phase I LP slacks;
2. maximize growth;
3. maximize ethanol production;
4. minimize glucose consumption;
5. minimize xylose consumption.

The upper bounds for glucose, xylose, and oxygen consumption have the following form:

$$\begin{aligned} v_g^{UB} &= v_{g, \max} \frac{g}{K_g + g} \frac{1}{1 + e/K_{ie}}, \\ v_z^{UB} &= v_{z, \max} \frac{z}{K_z + z} \frac{1}{1 + g/K_{ig}} \frac{1}{1 + e/K_{ie}}, \\ v_o^{UB} &= v_{o, \max} \frac{o}{K_o + o}, \end{aligned} \tag{7.37}$$

with parameters obtained from Table I in [51]. The metabolic network reconstruction used was iJR904 [107]. The work in [60] does not deal with the nonuniqueness nor the problems associated with the FBA LP becoming infeasible. In that case, the uptake kinetics are determined by the Michaelis-Menten expressions, and sensitivities can only be computed up to time 7 h. Figure 2 in [60] reports the solution to the DFBA simulation as well as the sensitivities with respect to glucose initial concentration and O_2 concentration. With the theory developed in this chapter, the use of the LLP allows using a penalty function which in turn allows the computation of sensitivities for any time length (10 hours in this example), and the uptake kinetics are computed using the LLP. Here, we compute the sensitivities with respect to biomass, glucose, xylose, oxygen, and ethanol initial concentrations (O_2 concentration is constant).

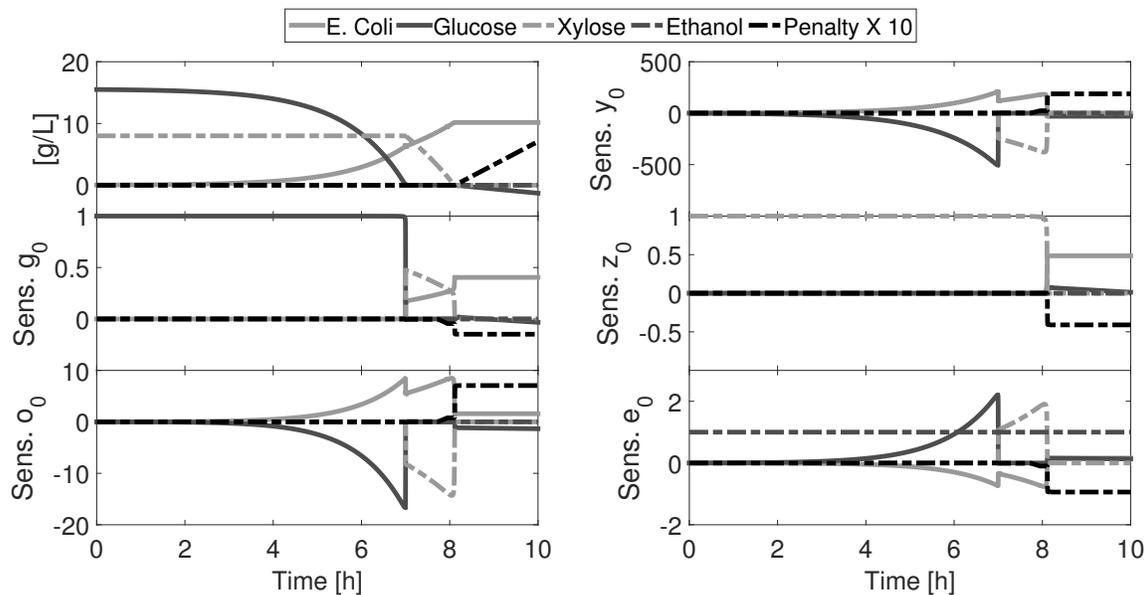


Figure 7-2: DFBA simulation and sensitivities for a batch process growing *E. coli* on glucose and xylose. The three plots on the left side coincide with the plots reported in Figure 2 of [60], only that this chapter is able to compute sensitivities past time 7 h and the DFBA simulation past time 8.2 hours. The plots on the right-hand side report the sensitivities for the initial concentration of biomass, xylose, and ethanol, respectively.

7.5.2 *E. coli*/yeast continuous cultivation system

The previous example was extended to consider yeast too. In addition, the oxygen concentration was made variable. The new dynamic system is the following:

$$\begin{aligned}
\dot{b}(t, \mathbf{p}) &= \mu_B(\mathbf{x}(t, \mathbf{p}), \mathbf{p})b(t, \mathbf{p}) - F_0b(t, \mathbf{p}), \\
\dot{y}(t, \mathbf{p}) &= \mu_Y(\mathbf{x}(t, \mathbf{p}), \mathbf{p})y(t, \mathbf{p}) - F_0y(t, \mathbf{p}), \\
\dot{g}(t, \mathbf{p}) &= -v_{g,B}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})b(t, \mathbf{p}) - v_{g,Y}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}), y(t, \mathbf{p}) + F_0(g_0 - g(\mathbf{x}(t, \mathbf{p}))), \\
\dot{z}(t, \mathbf{p}) &= -v_{z,B}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})b(t, \mathbf{p}) - v_{z,Y}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}), y(t, \mathbf{p}) + F_0(z_0 - z(\mathbf{x}(t, \mathbf{p}))), \\
\dot{e}(t, \mathbf{p}) &= v_{e,B}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})b(t, \mathbf{p}) + v_{e,Y}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})y(t, \mathbf{p}) - F_0e(t, \mathbf{p}), \\
\dot{o}(t, \mathbf{p}) &= -v_{o,B}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})b(t, \mathbf{p}) - v_{o,Y}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}), y(t, \mathbf{p}) \\
&\quad - F_0z(\mathbf{x}(t, \mathbf{p}) + k_{O_2}(o^* - o(t, \mathbf{p}))), \\
\dot{\alpha}(t, \mathbf{p}) &= \gamma_B(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) + \gamma_Y(\mathbf{x}(t, \mathbf{p}), \mathbf{p}),
\end{aligned}$$

where $\mathbf{x} \equiv [b, y, g, z, e, o, \alpha]$. The hierarchy of objectives for *E. coli* was modified by adding a sixth objective of minimizing oxygen consumption. This same hierarchy of objectives was used for yeast. The metabolic network used for yeast was iND750 [31]. Yeast uptake kinetics parameters were those reported in Table I in [51]. Using $F_0 = 0.5/h$, $g_0 = 10$ g/L, $z_0 = 5$ g/L and $k_{O_2} = 0.6/h$ the system was allowed to attain steady-state:

$$\mathbf{x}_{ss} = [1.65, 0, 3.84, 5.00, 0.014, 0.21, 0].$$

After reaching steady-state, it takes 5 seconds to compute the sensitivities in Figure 7-3.

7.6 Conclusions

This work represents an important step forward compared to the work reported in [60] from an optimization standpoint. Whereas sensitivities obtained by [60] are not

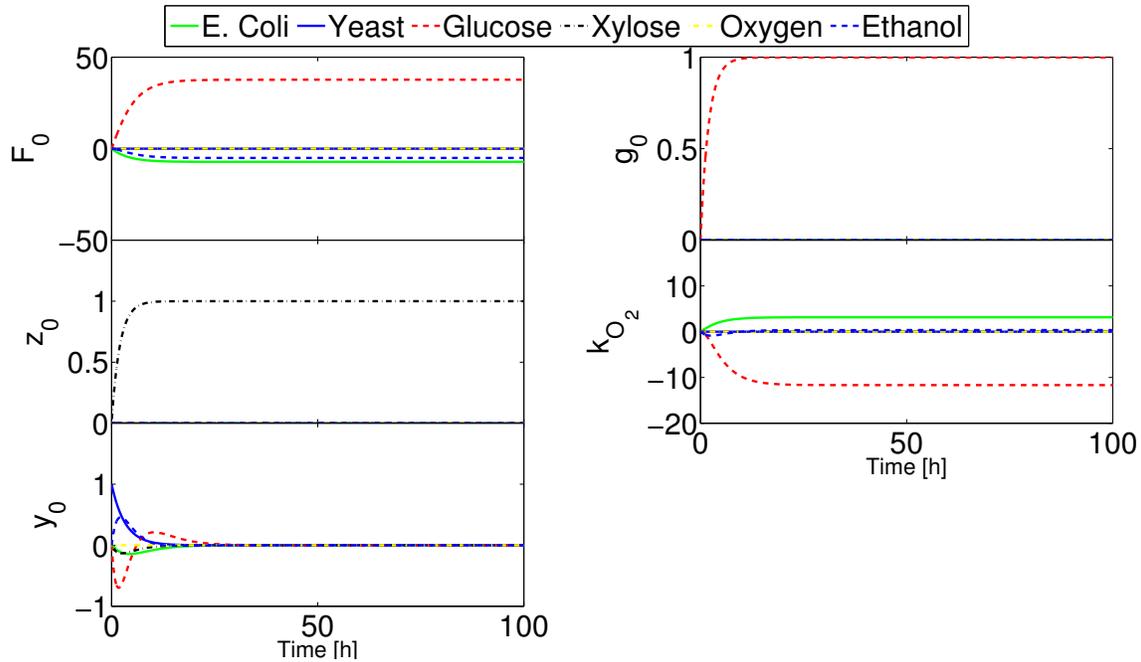


Figure 7-3: DFBA simulation and sensitivities for a continuous process involving *E. coli* and yeast. The plots give the sensitivities for an increase in dilution rate (top left), glucose feed concentration (top right), xylose feed concentration (middle left), mass transfer coefficient for O_2 (middle right), and yeast concentration (bottom left). Given that the steady state has no yeast present, the last plot essentially gives the sensitivity of the system to yeast invasion. It can be seen that yeast invasion is not stable in this system, and the system returns to the initial steady state.

amenable for optimization as sometimes they need to be truncated before the end time of the simulation, the work here allows the computation of sensitivities regardless of when the LLP becomes infeasible. This is critical to be able to solve optimization problems. Different classes of optimization problems are illustrated in the following chapter.

Chapter 8

Local Optimization of Dynamic Flux Balance Analysis Models

The optimization of dynamic flux balance analysis (DFBA) models enables the solution of the following kinds of problems:

1. Parameter estimation;
2. Optimal design of bioprocesses;
3. Optimal control of bioprocesses.

Given a closed set $Z \in \mathbb{R}^{n_p}$, general optimization problems of DFBA models have the following form:

$$\begin{aligned} \min_{\mathbf{p} \in Z} J(\mathbf{p}) &\equiv \varphi(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_0^{t_f} l(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt, \\ \text{s.t. } \mathbf{g}(\mathbf{p}) &\equiv \mathbb{R}(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_0^{t_f} \mathbf{s}(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt \leq 0. \end{aligned} \quad (8.1)$$

In general, J is a nonconvex and a nonsmooth function, and the constraints \mathbf{g} are nonsmooth and can describe a nonconvex feasible set. Therefore, this kind of optimization requires global optimization and nonsmooth strategies to find a global optimum. In this chapter, we shall focus only on performing local optimization of DFBA models while taking into account the nonsmooth nature of the problem.

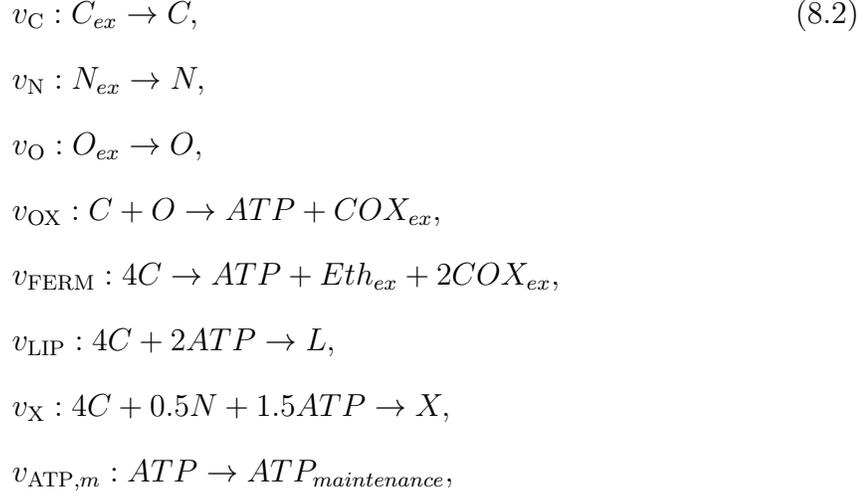
Global optimization requires convex and concave relaxations. Appendix C contains the derivation of these relaxations for lexicographic linear programs.

In this chapter we use derivative-based optimization. Given that this is a non-smooth optimization problem, generalized derivatives are needed. Here, we use the techniques in Chapters 6 and 7 to compute the sensitivities of DFBA systems. Then, we use nonsmooth optimization methods [80] to solve different classes of optimization problems. We also illustrate the use of the robust smooth optimizer IPOPT [139] with LD-derivatives. First, a toy metabolic model is used to show different kinds of optimization problems, and then an example using genome-scale metabolic networks is presented. The results presented in this chapter should be considered as preliminary.

8.1 Toy Metabolic Network

Here, we introduce a toy metabolic network. This metabolic network consumes a carbon source C, a nitrogen source N, and an oxygen source O to produce lipids L, ethanol E, biomass X, ATP and some oxidation product COX. This metabolic network is used for illustration purposes and is not meant to satisfy mass balances. However, it is supposed to reproduce the behavior of living organisms. In particular, E can only be produced in the absence of O, L can only be accumulated in the absence of N, there is a minimum ATP requirement, and the aerobic oxidation of C produces

more energy than the fermentation of C. The set of reactions is the following:



The subscript *ex* refers to extracellular metabolites. All these reactions are unidirectional (their lower bounds are equal to zero). Assume all reactions are in *mmol/gDW* except for reaction v_X in *gDW/gDW* and reaction v_{LIP} in *g/gDW*.

The simulations in this section were carried on MATLAB 7.12 running on Windows 10 Pro with a 64 bit operating system and a 3.20 GHz Intel(R) Xeon(R) processor.

8.1.1 Parameter Estimation Problem

Let us assume that the uptake kinetics are given by the following expressions:

$$\begin{aligned}
v_C^{UB}(\mathbf{x}) &= \max \left(0, v_{\max,C} \frac{C}{K_C + C} \frac{1}{1 + E/K_{iE}} \right), \\
v_N^{UB}(\mathbf{x}) &= \max \left(0, v_{\max,N} \frac{N}{K_N + N} \right), \\
v_O^{UB}(\mathbf{x}) &= \max \left(0, v_{\max,O} \frac{O}{K_O + O} \right),
\end{aligned} \tag{8.3}$$

where \mathbf{x} is a vector containing the extracellular concentration information. The following parameters are considered:

$$\mathbf{p} = \left[v_{\max,C} \quad K_C \quad v_{\max,N} \quad K_N \quad v_{\max,O} \quad K_O \quad K_{iE} \quad v_{\text{ATP},m} \right]^T.$$

To illustrate how a parameter estimation problem would work, ‘experimental data’ was generated using the following dynamic model for a batch reactor:

$$\begin{aligned} \dot{X}(t, \mathbf{p}, \mathbf{x}_0) &= \mu(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))X(t, \mathbf{p}, \mathbf{x}_0), \\ \dot{C}(t, \mathbf{p}, \mathbf{x}_0) &= -v_C(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))X(t, \mathbf{p}, \mathbf{x}_0), \\ \dot{N}(t, \mathbf{p}, \mathbf{x}_0) &= -v_N(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))X(t, \mathbf{p}, \mathbf{x}_0), \\ \dot{O}(t, \mathbf{p}, \mathbf{x}_0) &= -v_O(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))X(t, \mathbf{p}, \mathbf{x}_0), \\ \dot{L}(t, \mathbf{p}, \mathbf{x}_0) &= v_{\text{LIP}}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))X(t, \mathbf{p}, \mathbf{x}_0), \\ \dot{E}(t, \mathbf{p}, \mathbf{x}_0) &= v_{\text{FERM}}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))X(t, \mathbf{p}, \mathbf{x}_0), \\ C\dot{O}X(t, \mathbf{p}, \mathbf{x}_0) &= (v_{\text{OX}}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) + 2v_{\text{FERM}}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)))X(t, \mathbf{p}, \mathbf{x}_0), \\ \dot{\alpha}(t, \mathbf{p}, \mathbf{x}_0) &= v_\alpha, \\ \mathbf{x}(0, \mathbf{p}, \mathbf{x}_0) &= \mathbf{x}_0, \end{aligned} \tag{8.4}$$

where $\mathbf{x} = \left[X \quad C \quad N \quad O \quad L \quad E \quad COX \quad \alpha \right]^T$, \mathbf{x}_0 contains the initial conditions, α refers to the penalty state described in Chapter 3, and all reaction rates are obtained using the metabolic network 8.2 and flux balance analysis (FBA). Notice that the lipids are not considered to be part of the biomass in this formulation. Here, we use lexicographic optimization and the Phase I of the simplex algorithm to obtain a well-posed optimization problem. The stoichiometry matrix \mathbf{S} in Table 8.1 is obtained.

Table 8.1: Stoichiometry Matrix for Toy Metabolic Network.

	v_C	v_N	v_O	v_{OX}	v_{FERM}	v_{LIP}	v_X	$v_{\text{ATP},m}$
C	1	0	0	-1	-4	-4	-4	0
N	0	1	0	0	0	0	-0.5	0
O	0	0	1	-1	0	0	0	0
ATP	0	0	0	1	1	-2	-1.5	-1

The matrix has only four rows because there are only four intracellular metabolites in this network. The FBA problem corresponds to the following:

$$\begin{aligned} \max \quad & v_X, \\ \text{s.t.} \quad & \mathbf{S}\mathbf{v} = \mathbf{0}, \\ & \mathbf{0} \leq \mathbf{v} \leq \mathbf{v}^{UB}(\mathbf{x}). \end{aligned} \tag{8.5}$$

We proceed to transform this LP into standard form and then formulate a Phase I LP of (8.5). The upper bound inequality constraints are transformed into equality constraints using slack variables (rows 5 to 7 in Table 8.2) and this problem is converted into an LP that will be feasible for any right-hand side generated with (8.3) by adding a slack variable s_α to the ATP maintenance requirement). The technology matrix \mathbf{A} in Table 8.2 is obtained:

Table 8.2: Technology matrix for FBA problem in standard form.

	v_C	v_N	v_O	v_{OX}	v_{FERM}	v_{LIP}	v_X	$v_{ATP,m}$	s_1	s_2	s_3	s_α
C	1	0	0	-1	-4	-4	-4	0	0	0	0	0
N	0	1	0	0	0	0	-0.5	0	0	0	0	0
O	0	0	1	-1	0	0	0	0	0	0	0	0
ATP	0	0	0	1	1	-2	-1.5	-1	0	0	0	0
v_C^{UB}	0	0	0	0	0	0	0	0	1	0	0	0
v_N^{UB}	0	0	0	0	0	0	0	0	0	1	0	0
v_O^{UB}	0	0	0	0	0	0	0	0	0	0	1	0
$v_{ATP,m}$	0	0	0	0	0	0	0	0	0	0	0	1

The values v_C^{UB} , v_N^{UB} , v_O^{UB} , $v_{ATP,m}$ are all nonnegative, because of the max operator in 8.3. In this case, the slack variable s_α suffices to obtain a feasible LP at all times. Whenever there are not enough resources to satisfy the $v_{ATP,m}$ requirement, s_α takes a positive value while the bounds on uptake kinetics are still satisfied. In this way, when there are no remaining resources in the media, all metabolic fluxes will be equal to zero, which corresponds to a state where the cells are dead and is consistent with what would be observed experimentally. The hierarchical optimization problem has the following objectives:

Table 8.3: Hierarchy of objectives for the toy metabolic network in 8.2.

1	Minimize s_α
2	Maximize v_X
3	Maximize v_{LIP}
4	Maximize v_{FERM}
5	Minimize v_C
6	Minimize v_N
7	Minimize v_O
8	Minimize v_{COX}

The FBA problem in standard form is:

$$\begin{aligned} \text{lex min } & \mathbf{C}^T \mathbf{v} \\ \text{s.t. } & \mathbf{A} \mathbf{v} = \mathbf{b}(\mathbf{x}), \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{8.6}$$

where \mathbf{C} corresponds to the objectives in Table 8.3 and

$$\mathbf{b} = \left[0 \quad 0 \quad 0 \quad 0 \quad v_C^{UB} \quad v_N^{UB} \quad v_O^{UB} \quad v_{ATP,m} \right]^T.$$

To generate the data, the following parameter values were considered: $v_{\max,C} = 1.5$, $K_C = 0.05$, $v_{\max,N} = 0.25$, $K_N = 0.5$, $v_{\max,O} = 2$, $K_O = 1.2$, $K_{iE} = 15$, and $v_{ATP,m} = 0.18$. These shall be considered the base parameter values. The data was generated using MATLAB's stiff ODE integrator `ode15s`, with relative and absolute integration tolerances of 1×10^{-9} and LP solver Gurobi [50] with optimality and feasibility tolerances of 1×10^{-9} . The exact values generated were the following:

Table 8.4: Simulation data for Toy Metabolic Network.

Time [h]	X	C	N	O	L	E	COX
0	0.01	15	0.3	1	0	0	0
10	0.0628	14.567	0.2736	0.8384	0.0151	0	0.1616
20	0.2958	12.215	0.1571	0.1339	0.0985	0.0953	1.057
30	0.5675	5.733	0.0212	7.68×10^{-5}	0.173	1.336	3.672
40	0.6052	0	0.002401	1.24×10^{-8}	0.348	2.557	6.114

Notice that the first row corresponds to known values (what is loaded into the batch reactor at time zero). Although the penalty α takes positive values, it is not

reported as this value cannot be measured experimentally. Considering that $\alpha > 0$ starting at $t \approx 37.5$, the data generated is considered to be valid for all 40 hours given that the way the penalty is formulated, no extracellular concentrations become negative.

In reality, experimental data contains noise. Here, we added noise obtaining random numbers from a normal distribution centered at zero and with standard deviations reported on the top row of Table 8.5. Then, these numbers were added to the nonzero values. Very small numbers were considered to be equal to zeroes. Figures 8-1 and 8-2 show the simulated values compared to the noisy experimental data.

Table 8.5: Experimental data for Toy Metabolic Network.

Time [h]	X	C	N	O	L	E	COX
STDEV	0.02	0.4	0.01	0.05	0.01	0.1	0.1
10	0.0654	14.502	0.2648	0.8288	0.0044	0	0.1355
20	0.3061	12.157	0.1523	0.1202	0.1145	0.0509	1.1011
30	0.5727	5.520	0.0141	0.0766	0.1853	1.3206	3.7116
40	0.5864	0	0	0	0.3454	2.5847	5.989

Assuming the parameter values are unknown, a minimization of squared errors problem can be formulated as $\min \sum_{i=1}^{28} e_i^2$ where $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$, $\hat{\mathbf{y}}$ corresponds to experimental values and \mathbf{y} corresponds to predicted values. Here, all experimental measurements are given the same weight. Some weighting factors can be added in case some experimental values are considered to be more important than others.

To solve this optimization problem, function and gradient evaluations are needed. The simulations required to compute the function and the gradient were carried on using MATLAB's ODE integrator `ode15s` with absolute and relative tolerances of 1×10^{-6} and LP solver Gurobi with optimality and feasibility tolerances of 1×10^{-7} . Gradients were computed using the techniques exposed in Chapters 6 and 7. Under these conditions, the base parameter values give a function value of 0.0829 because of the noise introduced in the data.

This optimization problem was solved using nonsmooth optimizer `Solvopt` [80]. The problem was formulated as an unconstrained minimization problem. To ensure all constants were nonnegative, the absolute values of the optimization variables were

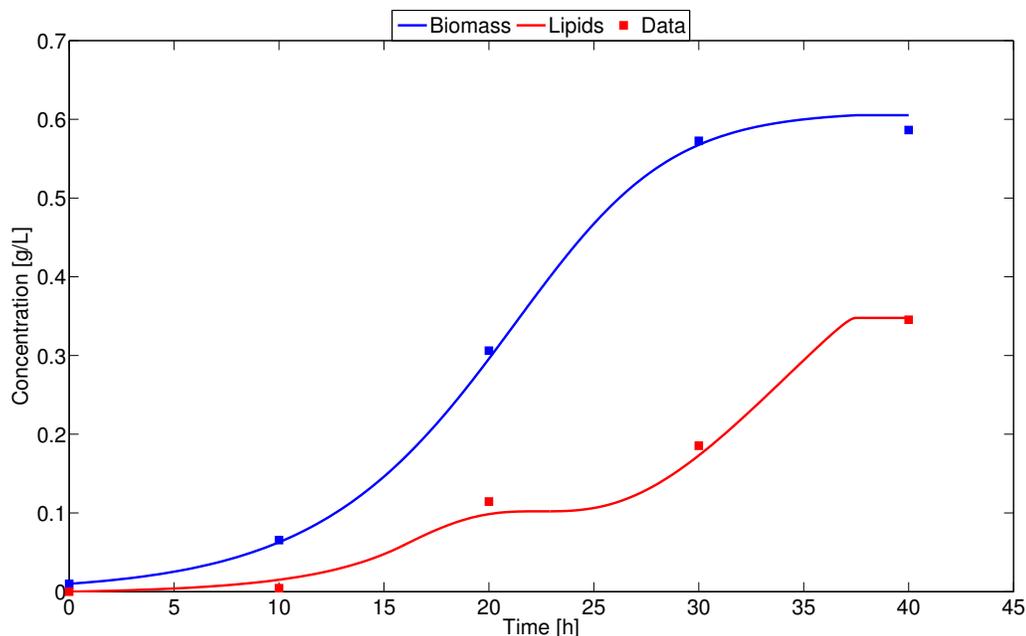


Figure 8-1: Simulated and experimental data for biomass and lipids for a batch experiment using the toy metabolic network. Solid lines provide the simulation results using the base parameter values whereas squared markers refer to noisy experimental data.

used to run the simulations. The tolerances used in Solvopt were 1×10^{-4} for the relative error of the argument in terms of the infinity norm and for the relative error of the function value. A minimum value of 0.001 was chosen for all parameters and was enforced using a max statement because some parameter values approached zero causing numerical difficulties. Using a start point of $\mathbf{p}_0 = [2, 2, 0.5, 5, 3, 1, 10, 0.1]^T$, Solvopt takes 188 seconds to terminate. It performs 531 function evaluations and 157 gradient evaluations and finds point

$$\mathbf{p}^* = [1.5628, 0.001, 0.1946, 0.3338, 29000, 17000, 7.945, 0.1716]^T.$$

It exits with the following termination warning: “Result may not provide the optimum. The function is flat at the optimum.” Despite not having a normal termination, the objective improves from 272.3 to 0.0228. In this specific example, using a finite differencing scheme with $\delta = 1E-6$ takes 152 seconds and 129 iterations, 438 function

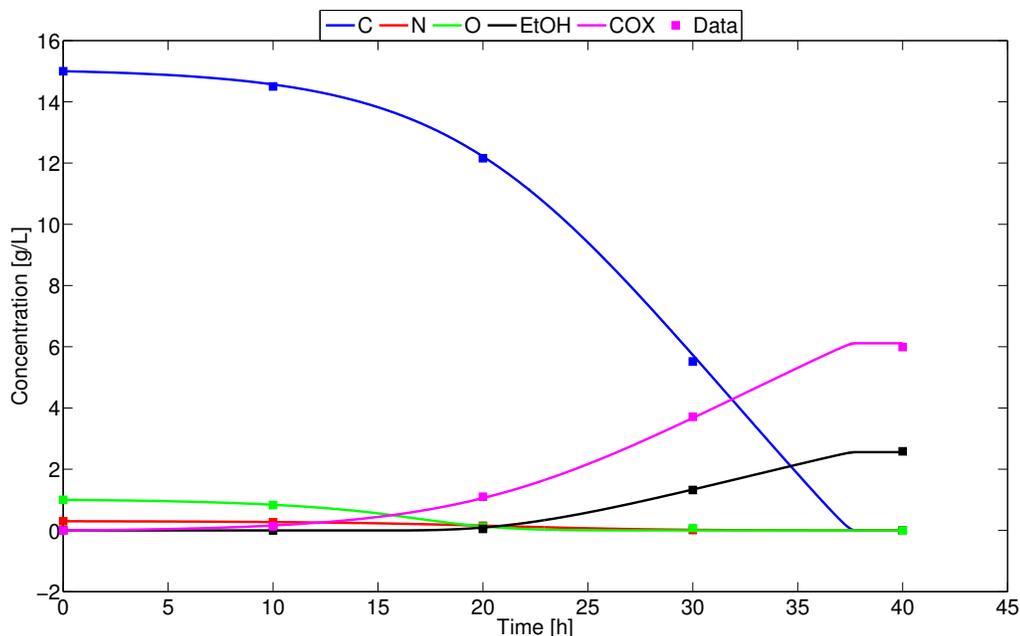


Figure 8-2: Simulated and experimental data for substrates and products for a batch experiment using the toy metabolic network. Solid lines provide the simulation results using the base parameter values whereas squared markers refer to noisy experimental data.

evaluations and 130 gradient evaluations to find point

$$\mathbf{p}^* = [1.5074, 0.3293, 0.3217, 0.7046, 5.607, 3.1952, 3100, 0.1685]^T,$$

with an objective of 0.0237. Solvopt has a normal termination. Table 8.6 presents a comparison of the different parameter vectors. Figures 8-3 and 8-4 show the drastic improvement that can be attained using optimization.

Using finite differences can be risky. In particular, if the gradient of the objective function is approximated using finite differences at \mathbf{p}_0 , the following results are obtained:

1. $\delta = 1 \times 10^{-5} : \nabla f \approx [-50.62, 6.38, -260.96, 24.66, 0.062, -0.15, -0.0012, -15.71]$.
2. $\delta = 1 \times 10^{-6} : \nabla f \approx [-50.59, 6.40, -261.15, 24.65, 0.082, -0.18, -0.0012, -15.69]$.
3. $\delta = 1 \times 10^{-7} : \nabla f \approx [-50.40, 4.23, -260.74, 22.50, 0.060, -0.21, -0.0012, -15.50]$.

Table 8.6: Comparison of optimization results with initial and base points.

	$\mathbf{p}^*(LD)$	$\mathbf{p}^*(FD)$	\mathbf{p}_0	\mathbf{P}_{base}
$v_{max,C}$	1.563	1.5074	2	1.5
K_C	0.001	0.3293	2	0.05
$v_{max,N}$	0.1946	0.3217	0.5	0.25
K_N	0.3338	0.7046	5	0.5
$v_{max,O}$	29000	5.607	3	2
K_O	17000	3.1952	1	1.2
$K_{i,E}$	7.945	3100	10	15
$v_{ATP,m}$	0.1716	0.1685	0.1	0.18
$\sum_i e_i^2$	0.0228	0.0237	272.3	0.0829
Time [s]	188	152		

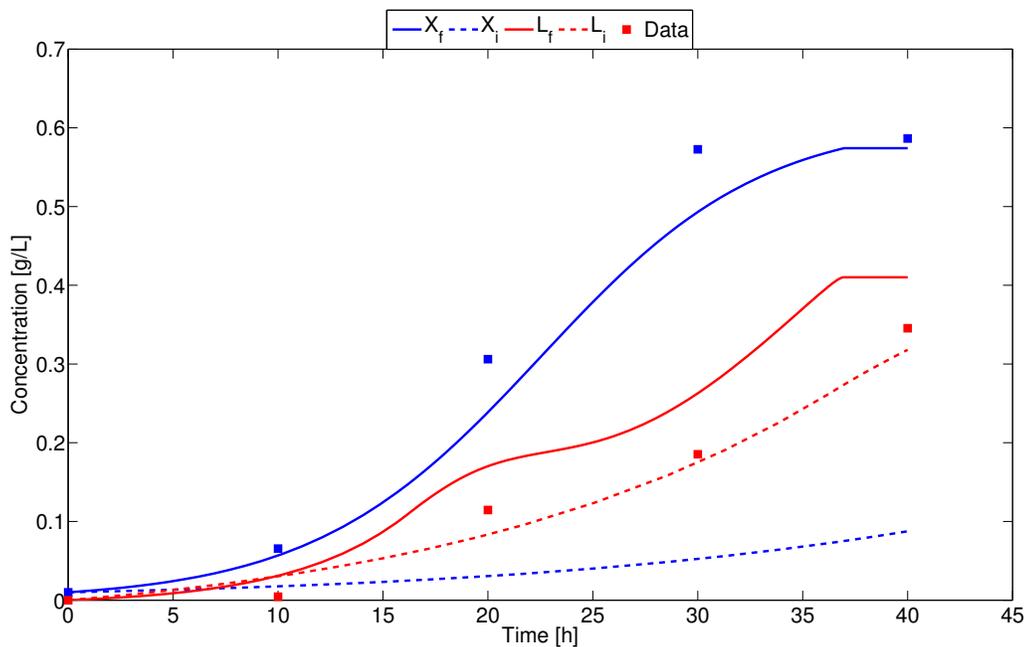


Figure 8-3: Simulated and experimental data for biomass and lipids for a batch experiment using the toy metabolic network. Solid lines provide the simulation results using the optimized parameter values obtained with LD-derivatives (subscript f), and dashed lines were simulated using the optimization initial point (subscript i).

It has to be noted that not all parameters were estimated accurately. There are several explanations for this. First, the noise may have shifted in a significant manner the optimal point from the base parameter values to some other point in the parameter space. Second, the measurements are in different scales. The largest measurements are those used for C , therefore, the optimizer does quite well fitting the C curve to

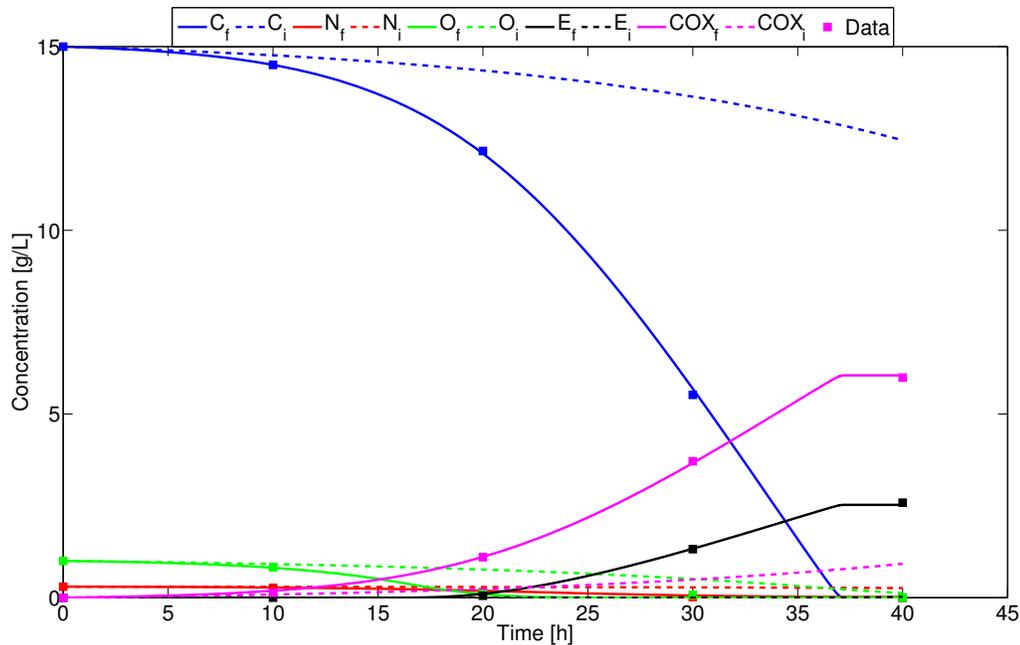


Figure 8-4: Simulated and experimental data for substrates and products for a batch experiment using the toy metabolic network. Solid lines provide the simulation results using the optimized parameter values obtained with LD-derivatives (subscript f), and dashed lines were simulated using the optimization initial point (subscript i).

the experimental value. Weights can be added as prefactors to the errors to normalize the different quantities. Finally, given that this problem is nonconvex, the optimizer may have terminated near a local minimum. To help the minimizer find the real parameter values, several batch experiments using different initial conditions can be performed. Nevertheless, there is a very meaningful improvement in function value that can be observed in Figures 8-3 and 8-4.

The strategy of using weights as prefactors was implemented. For each measured variable, the reciprocal of the largest data point was used as a prefactor. For example, a value of $1/0.5864$ premultiplied all measurements and all predictions of X (see Table 8.5).

For this optimization problem, the same start point was used. After 171 seconds, 157 iterations, 516 function evaluations and 158 gradient evaluations, a solution point

of

$$\mathbf{p}^* = [1.5293, 0.0933, 0.2325, 0.4376, 20000, 13200, 7.2904, 0.1693]^T$$

was found. Solvopt is able to finish with no warnings. The summary of results is presented in Table 8.7.

Table 8.7: Comparison of optimization results with initial and base points when weights are used.

	\mathbf{p}^*	\mathbf{p}_0	\mathbf{p}_{base}
$v_{\max,C}$	1.5293	2	1.5
K_C	0.0933	2	0.05
$v_{\max,N}$	0.2325	1	0.25
K_N	0.4376	5	0.5
$v_{\max,O}$	20000	3	2
K_O	13200	1	1.2
$K_{i,E}$	7.2904	10	15
$v_{ATP,m}$	0.1693	0.1	0.18
$\sum_i w_i e_i^2$	0.0137	8.2711	0.0183

Again, the $v_{\max,O}$ and K_O parameters are not estimated accurately. This can be explained by the fact that the data points may have been taken when either O is not limiting or when it is not present in the system. For the following computational experiments, the base parameter values will be assumed to be known.

8.1.2 Optimal design of a batch process

In this section, the optimal design problem of a batch reactor is formulated. This reactor can be modeled using the dynamical system (8.4), only that now $\mathbf{x}_0(\mathbf{p}) = [0.01, p_1, p_2, p_3, 0, 0, 0, 0]$ and $t_f = p_4$ (the parameters vector is $\mathbf{p} = [C_0, N_0, O_0, t_f]$). The profit function is given by

$$P(\mathbf{p}) = 50L(t_f, \mathbf{p}) + 10E(t_f, \mathbf{p}) - 2p_1 - 5p_2 - 0.5p_3^2 - p_4,$$

where the last term includes a time dependency. The total time of the batch process is also a parameter. Remember that by the way the phase I LP was formulated,

when the FBA LP becomes infeasible, growth remains at zero and the mass balances are still satisfied. Since total batch time is now a parameter that incurs cost, any system where $\alpha(t_f, \mathbf{p}) > 0$ is suboptimal. The dynamical sensitivities of this system can be computed using the theory presented in Chapters 6 and 7. A summary of the results can be found in Table 8.8. The results in these subsection were computed with relative and absolute integration tolerances of 1×10^{-7} and LP feasibility and optimality tolerances of 1×10^{-7} .

Table 8.8: Summary of results for the optimal design of a batch process using P .

	LD-derivatives	FD ($\delta = 10^{-6}$)
\mathbf{p}_0	[20, 0.1, 1, 40]	[20, 0.1, 1, 40]
\mathbf{p}^*	[82.03, 0.606, 11.58, 72.97]	[79.91, 0.596, 11.42, 72.00]
$P(\mathbf{p}_0)[\frac{\$}{(L \times batch)}]$	-65.03	-65.03
$P(\mathbf{p}^*)[\frac{\$}{(L \times batch)}]$	41.00	40.96
Time [s]	41	69
# Iterations	44	70
Warnings	Normal termination.	Normal termination.
\mathbf{p}_0	[1, 1, 1, 1]	[1, 1, 1, 1]
\mathbf{p}^*	[0, 0, 0, 0]	[0, 0, 0.0003, 0]
$P(\mathbf{p}_0)[\frac{\$}{(L \times batch)}]$	-8.5	-8.5
$P(\mathbf{p}^*)[\frac{\$}{(L \times batch)}]$	8.0×10^{-10}	1.8×10^{-6}
Time [s]	2.8	7
# Iterations	41	79
Warnings	Normal termination. Trivial solution: no process is carried out.	Normal termination. Same as LD-derivatives.
\mathbf{p}_0	[40, 2, 1, 80]	[40, 2, 1, 80]
\mathbf{p}^*	[82.03, 0.606, 11.58, 72.97]	[86.19, 0.62, 11.92, 74.86]
$P(\mathbf{p}_0)[\frac{\$}{(L \times batch)}]$	-92	-92
$P(\mathbf{p}^*)[\frac{\$}{(L \times batch)}]$	41.00	40.81
Time [s]	39	40
# Iterations	41	41
Warnings	Normal Termination.	Normal termination.

It is pretty clear in this example that LD-derivatives perform better than finite differences. In all cases, LD-derivatives take less time, find a better objective and in two instances take less iterations. Figures 8-5 and 8-6 show concentration profiles for the optimal parameters of the batch process.

Another interesting problem could be to maximize profit in a per hour basis instead

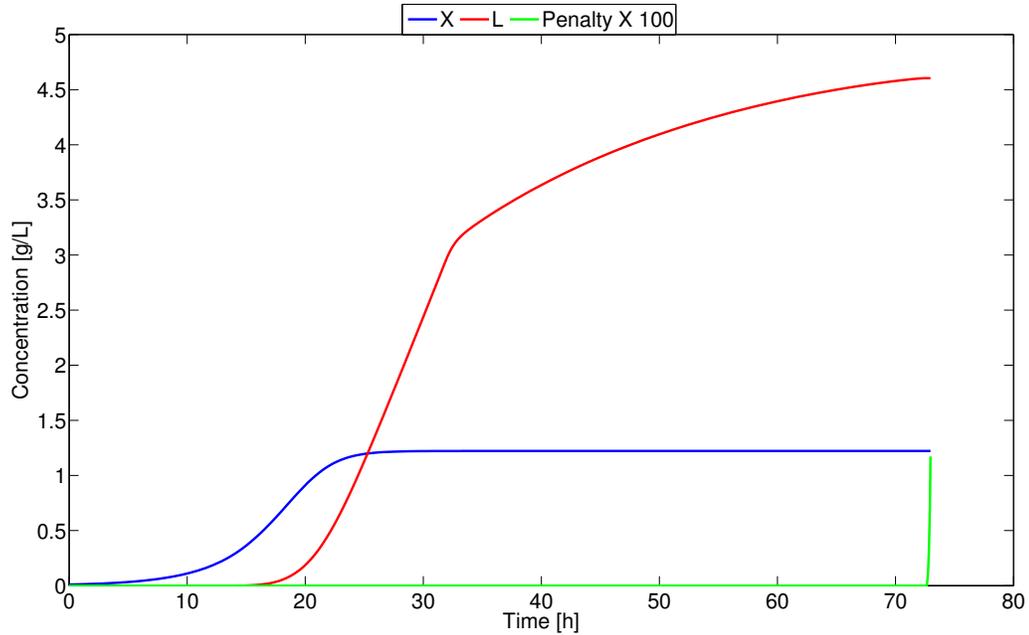


Figure 8-5: Biomass and lipids concentrations for optimal batch parameters. It can be seen that the optimizer reduces the time until penalty goes almost to zero.

of a per batch process. Therefore, the new objective function would be:

$$\hat{P}(\mathbf{p}) = \frac{50L(t_f, \mathbf{p}) + 10E(t_f, \mathbf{p}) - 2p_1 - 5p_2 - 0.5p_3^2 - p_4}{p_4}.$$

A summary of the results for maximizing \hat{P} can be found in Table 8.9. Again, LD-derivatives perform better than finite differences in all criteria: objective function value, number of iterations, and total time of optimization.

Figures 8-7 and 8-8 show the concentration profiles for the optimal parameters of the batch process when maximizing profit in a per hour basis. The optimal point found by Solvopt for maximizing \hat{P} has a suboptimal value of $\$36.29/(L \times batch)$ in P . Meanwhile the optimal point when maximizing P has a suboptimal value of $\$0.562/(L \times h)$ in \hat{P} . The optimizer chooses a shorter batch time (56.43 h vs. 72.97 h) to maximize profit in a per hour basis compared to a per batch basis. In this case, the optimizer still adjusts the inputs such that all resources are fully utilized and it stops just as the penalty function starts increasing, which can be related to

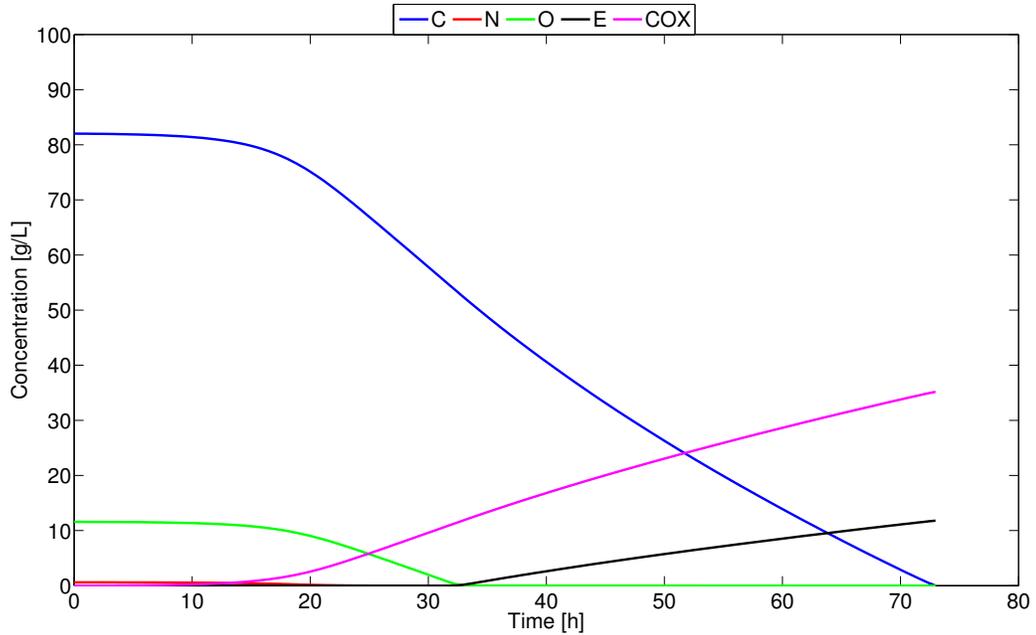


Figure 8-6: Substrate and product concentrations for optimal batch parameters. It can be seen that all substrates go to zero which implies that resources are fully utilized.

the microorganism dying.

8.1.3 Optimal Design of a Continuous Process Operating at Steady State

Consider again the toy metabolic network and assume that the base parameter values are the true values for the uptake kinetics expressions in Eq. (8.3). Now, consider that the process operates as a continuous stirred-tank reactor (CSTR) and that the design variables include the feed/output flowrate F , and the feed concentrations F_C , F_O , and F_N . It is desired that the system operates at steady state. Assume that for a given set of conditions, the profit of the system is given by:

$$P(\mathbf{x}_{ss}) = F (10E_{ss} + 50L_{ss} - 2F_C - 5F_N - 0.5F_O^2), \quad (8.7)$$

with units of \$/h.

Table 8.9: Summary of results for the optimal design of a batch process using \hat{P} .

	LD-derivatives	FD ($\delta = 10^{-6}$)
\mathbf{p}_0	[40, 2, 1, 80]	[40, 2, 1, 80]
\mathbf{p}^*	[68.09, 0.674, 10.95, 56.43]	[70.12, 0.6867, 11.06, 57.25]
$\hat{P}(\mathbf{p}_0)[\frac{\$}{(L \times h)}]$	-1.15	-1.15
$\hat{P}(\mathbf{p}^*)[\frac{\$}{(L \times h)}]$	0.643	0.642
Time [s]	35	74
# Iterations	38	73
Warnings	Normal termination.	Normal termination.
\mathbf{p}_0	[1, 1, 1, 1]	[1, 1, 1, 1]
\mathbf{p}^*	[68.09, 0.674, 10.95, 56.43]	[70.5, 0.689, 11.12, 57.39]
$\hat{P}(\mathbf{p}_0)[\frac{\$}{(L \times h)}]$	-8.5	-8.5
$\hat{P}(\mathbf{p}^*)[\frac{\$}{(L \times h)}]$	0.643	0.642
Time [s]	40	88
# Iterations	70	105
Warnings	Normal termination.	Normal termination.
\mathbf{p}_0	[20, 0.1, 1, 40]	[20, 0.1, 1, 40]
\mathbf{p}^*	[68.09, 0.674, 10.95, 56.43]	[65.24, 0.655, 10.78, 55.37]
$\hat{P}(\mathbf{p}_0)[\frac{\$}{(L \times h)}]$	-1.6256	-1.6256
$\hat{P}(\mathbf{p}^*)[\frac{\$}{(L \times h)}]$	0.643	0.642
Time [s]	39	151
# Iterations	42	154
Warnings	Normal termination.	Normal termination.

Now, the dynamic system has the following form:

$$\begin{aligned}
 \dot{X}(t, \mathbf{p}, \mathbf{x}_0) &= \left(\mu(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) - \frac{F(\mathbf{p})}{V} \right) X(t, \mathbf{p}, \mathbf{x}_0), \\
 \dot{C}(t, \mathbf{p}, \mathbf{x}_0) &= \frac{F(\mathbf{p})}{V} (F_C(\mathbf{p}) - C(t, \mathbf{p}, \mathbf{x}_0)) - v_C(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) X(t, \mathbf{p}, \mathbf{x}_0), \\
 \dot{N}(t, \mathbf{p}, \mathbf{x}_0) &= \frac{F(\mathbf{p})}{V} (F_N(\mathbf{p}) - N(t, \mathbf{p}, \mathbf{x}_0)) - v_N(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) X(t, \mathbf{p}, \mathbf{x}_0), \\
 \dot{O}(t, \mathbf{p}, \mathbf{x}_0) &= \frac{F(\mathbf{p})}{V} (F_O(\mathbf{p}) - O(t, \mathbf{p}, \mathbf{x}_0)) - v_O(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) X(t, \mathbf{p}, \mathbf{x}_0), \\
 \dot{L}(t, \mathbf{p}, \mathbf{x}_0) &= -\frac{F(\mathbf{p})}{V} L(t, \mathbf{p}, \mathbf{x}_0) + v_{LIP}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) X(t, \mathbf{p}, \mathbf{x}_0), \\
 \dot{E}(t, \mathbf{p}, \mathbf{x}_0) &= -\frac{F(\mathbf{p})}{V} E(t, \mathbf{p}, \mathbf{x}_0) + v_{FERM}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) X(t, \mathbf{p}, \mathbf{x}_0), \\
 C\dot{O}X(t, \mathbf{p}, \mathbf{x}_0) &= -\frac{F(\mathbf{p})}{V} (COX(t, \mathbf{p}, \mathbf{x}_0)) + (v_{OX}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) \\
 &\quad + 2v_{FERM}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))) X(t, \mathbf{p}, \mathbf{x}_0), \\
 \dot{\alpha}(t, \mathbf{p}, \mathbf{x}_0) &= v_\alpha X(t, \mathbf{p}, \mathbf{x}_0), \\
 \mathbf{x}(0, \mathbf{p}, \mathbf{x}_0) &= \mathbf{x}_0, \quad [F(\mathbf{p}), F_C(\mathbf{p}), F_N(\mathbf{p}), F_O(\mathbf{p})] = \mathbf{p}.
 \end{aligned} \tag{8.8}$$

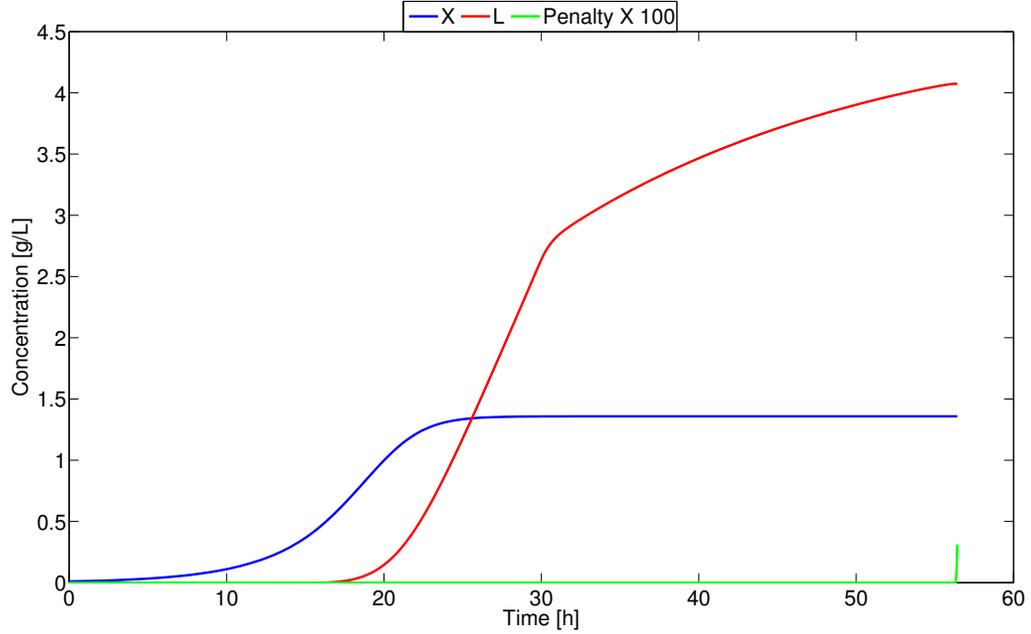


Figure 8-7: Biomass and lipids concentrations for optimal batch parameters. It can be seen that the optimizer reduces the time until penalty goes almost to zero.

Two main optimization strategies can be used. In the first strategy, the optimization problem has only four optimization variables $\mathbf{p} = [F, F_C, F_N, F_O]^T$ where F is in $1/h$ and F_C, F_N , and F_O are all in mol/L . In this case, the optimization problem has the following form:

$$\begin{aligned} \min_{\mathbf{p}} -P(\mathbf{p}) &= -F(\mathbf{p}) (10E_{ss}(\mathbf{p}) + 50L_{ss}(\mathbf{p}) - 2F_C(\mathbf{p}) - 5F_N(\mathbf{p}) - 0.5F_O(\mathbf{p})^2) \\ \text{s.t. } [F(\mathbf{p}), F_C(\mathbf{p}), F_N(\mathbf{p}), F_O(\mathbf{p})] &= \mathbf{p}. \end{aligned}$$

To be able to solve this problem, the LD-derivative $\mathbf{x}'(\mathbf{p}; \mathbf{M})$ needs to be computed. The relationship between \mathbf{x} and \mathbf{p} is given by the implicit function resulting from the steady state conditions $\dot{\mathbf{x}}(\mathbf{x}, \mathbf{p}) = \mathbf{f}(\mathbf{x}, \mathbf{p}) = \mathbf{0}$; therefore, the implicit function theorem for LD-derivatives is needed.

Theorem 8.1.1. (Theorem 2 in [75]). Let $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^m$ be open and $\mathbf{g} : X \times Y \rightarrow \mathbb{R}^m$ be lexicographically smooth (l -smooth) at $(\mathbf{x}^0, \mathbf{y}^0) \in X \times Y$.

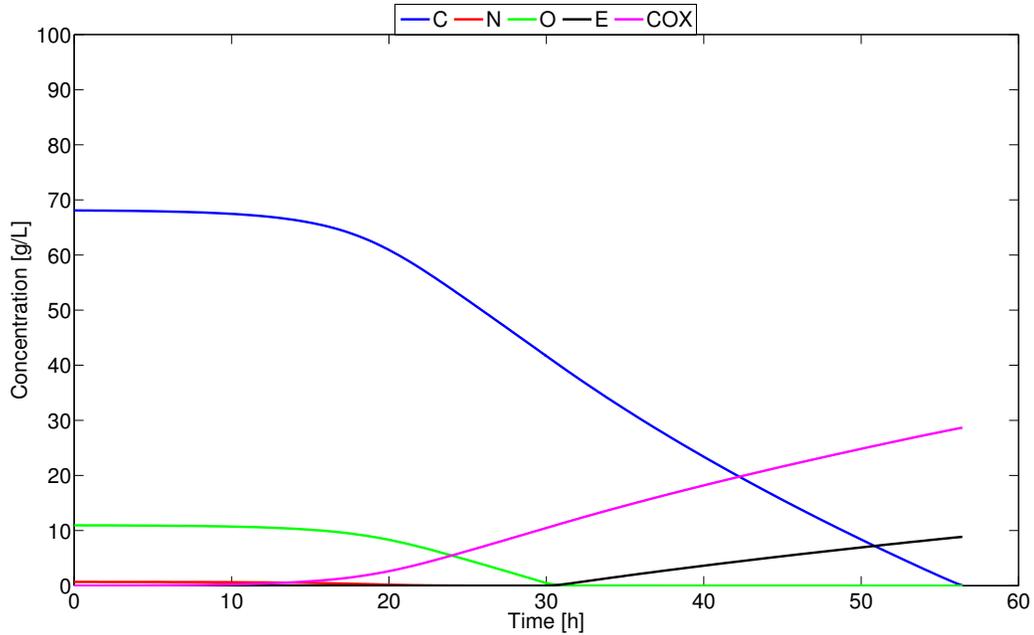


Figure 8-8: Substrate and product concentrations for optimal batch parameters. It can be seen that all substrates go to zero which implies that resources are fully utilized.

Suppose that $\mathbf{g}(\mathbf{x}^0, \mathbf{y}^0) = \mathbf{0}_m$ and, in addition, the auxiliary mapping $\mathbf{f} : X \times Y \rightarrow \mathbb{R}^n \times \mathbb{R}^m : (\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{x}, \mathbf{g}(\mathbf{x}, \mathbf{y}))$ is a Lipschitz homeomorphism at $(\mathbf{x}^0, \mathbf{y}^0)$. Then, there exists a neighborhood $N(\mathbf{x}^0) \subset X$ and a function $\mathbf{r} : N(\mathbf{x}^0) \rightarrow \mathbb{R}^m$ that is Lipschitz continuous on $N(\mathbf{x}^0)$ such that, for each $\mathbf{x} \in N(\mathbf{x}^0)$, $(\mathbf{x}, \mathbf{r}(\mathbf{x}))$ is the unique vector in a neighborhood of $(\mathbf{x}^0, \mathbf{y}^0)$ satisfying $\mathbf{g}(\mathbf{x}, \mathbf{r}(\mathbf{x})) = \mathbf{0}_m$. Moreover, \mathbf{r} is l -smooth at \mathbf{x}^0 , and for any $k \in \mathbb{N}$ and any $\mathbf{M} \in \mathbb{R}^{n \times k}$, the LD-derivative $\mathbf{r}'(\mathbf{x}^0; \mathbf{M})$ is the unique solution $\mathbf{N} \in \mathbb{R}^{m \times k}$ of the equation system

$$\mathbf{g}'(\mathbf{x}^0, \mathbf{y}^0; (\mathbf{M}, \mathbf{N})) = \mathbf{0}_{m \times k}. \quad (8.9)$$

The LD-derivative $\mathbf{x}'(\mathbf{p}; \mathbf{M})$ can be computed as an equation-solving problem using the results in Lemma 3.5 of [124] or by running the dynamical sensitivities in Chapter 7 until both, the variables and the sensitivities attain steady state. Framing the problem as an equation solving problem has the disadvantage that the washout solution is very attractive (see Example 6.5.3). Alternatively, the dynamic simulation

can be used to find the non trivial solution, and then the equation solving procedure can be used to find the LD-derivatives. The LD-derivatives for the implicit function can also be found by letting the sensitivities run for a long enough period of time. From Corollary 4.3 in [72], the sensitivity of an ODE system

$$\dot{\mathbf{y}}(t, \mathbf{c}) = \mathbf{f}(t, \mathbf{y}(t, \mathbf{c})), \quad \mathbf{y}(0, \mathbf{c}) = \mathbf{c}, \quad (8.10)$$

with \mathbf{f} being l -smooth is the unique solution of the following ODE system:

$$\dot{\mathbf{A}}(t) = \mathbf{f}'(\mathbf{y}(t, \mathbf{c}); \mathbf{A}(t)), \quad \mathbf{A}(0) = \mathbf{K}. \quad (8.11)$$

The steady state condition implies that $\mathbf{f}(t, \mathbf{y}(t, \mathbf{p})) = \mathbf{0}$. In our dynamic case, let $\mathbf{y} = [\mathbf{x}; \mathbf{p}]$ and $\mathbf{K} = [\mathbf{N}; \mathbf{M}]$, and let the last n_p components of \mathbf{f} be always equal to zero. At steady state, the solution of (8.10) will be $\mathbf{y}_{ss} = [\mathbf{x}_{ss}; \mathbf{p}]$, and the solution of 8.11 will be $\mathbf{A}_{ss} = [\mathbf{N}_{ss}; \mathbf{M}]$ and it will satisfy $\mathbf{f}'(\mathbf{x}_{ss}, \mathbf{p}; \mathbf{N}_{ss}, \mathbf{M}) = \mathbf{0}$ which is consistent with (8.9). The following Example illustrates this.

Example 8.1.1. Consider a CSTR with a reaction $A \rightarrow B$ and a rate constant $k = 1/h$ such that $r_A = -kA$, $F = 1L/h$ and $V = 1L$. Let the parameter be the feed concentration of A , $A_0 = p = 1 \text{ mol/L}$. Then,

$$\dot{A} = \frac{F}{V}(A_0 - A) - kA, \quad \dot{B} = -\frac{F}{V}B + kA. \quad (8.12)$$

The steady state of this system corresponds to $A_{ss} = B_{ss} = 1/2$.

Using the implicit function theorem, we can obtain the sensitivities $\mathbf{s}'(p; 1)$ where $\mathbf{s} = [A, B]$. Just use the quantities:

$$\frac{\partial \mathbf{f}}{\partial p} = \begin{bmatrix} \frac{F}{V} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{\partial \mathbf{f}}{\partial \mathbf{s}} = \begin{bmatrix} -\frac{F}{V} - k & 0 \\ k & -\frac{F}{V} \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 1 & -1 \end{bmatrix}, \quad (8.13)$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial p} = -\frac{\partial \mathbf{f}}{\partial p}, \quad \begin{bmatrix} -2 & 0 \\ 1 & -1 \end{bmatrix} \frac{\partial \mathbf{s}}{\partial p} = -\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{\partial \mathbf{s}}{\partial p} = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}. \quad (8.14)$$

Another way of computing these sensitivities is using the following dynamical system:

$$\dot{A} = p - 2A, \dot{B} = -B + A, A(0) = A_{ss}(p) = 0.5, B(0) = B_{ss}(p) = 0.5, \quad (8.15)$$

$$\dot{\mathbf{S}} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{x}_0} \end{bmatrix} \mathbf{S}(t) = \begin{bmatrix} -2 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{S}(t), \mathbf{S}(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (8.16)$$

which is equivalent to integrating the system:

$$\dot{S}_1 = -2S_1 + 1, \dot{S}_2 = S_1 - S_2, S_1(0) = S_2(0) = 0. \quad (8.17)$$

This system has a solution of $S_1(t) = -0.5e^{-2t} + 0.5$ and $S_2(t) = -e^{-t} + 0.5e^{-2t} + 0.5$ and when $t \rightarrow \infty$, $S_1(\infty) = S_2(\infty) = 0.5$ which correspond to the sensitivities computed using the implicit function theorem.

This first strategy for optimizing a continuous process is suitable for interior point methods as all iterates generated are feasible in the optimization problem. Alternatively, the optimization problem can be formulated in such a way that both parameters and states are optimization variables and the steady state conditions are added as constraints:

$$\min_{\mathbf{x}, \mathbf{p}} -P(\mathbf{x}) = -F(10E + 50L - 2F_C - 5F_N - 0.5F_O^2),$$

$$\text{s.t. } \dot{\mathbf{x}}(\mathbf{x}, \mathbf{p}) = \mathbf{f}(\mathbf{x}, \mathbf{p}) = \mathbf{0},$$

$$[F(\mathbf{p}), F_C(\mathbf{p}), F_N(\mathbf{p}), F_O(\mathbf{p})] = \mathbf{p}, L = x_5, E = x_6.$$

When the problem is formulated in this way, the implicit function theorem for LD-derivatives is not required. However, optimization iterates are not guaranteed to be feasible except at the solution where all constraints will be satisfied.

The results in these subsection were computed with relative and absolute integration tolerances of 1×10^{-7} and LP feasibility and optimality tolerances of $1 \times$

10^{-8} . First, the first strategy is used to optimize the system. To avoid numerical problems associated with F being too small, a minimum value of $0.001/h$ was enforced using a max function. To calculate the sensitivities of the implicit function, the sensitivities ODE system was allowed to reach steady state. With a start point of $\mathbf{p} = [0.01, 10, 1, 2]$, the optimizer takes 75 iterations, 322 function evaluations, and 76 gradient evaluations in 65 seconds to find the solution point of $\mathbf{p}_0 = [0.0496, 22.2558, 0.4170, 8.9434]$ with no termination warnings. In this way the objective improves from 0.0884 to -1.6351 with profit for this continuous system going from $-\$0.0884/(L \times h)$ to $\$1.6351/(L \times h)$.

Table 8.10: Comparison of the optimal result with start point for continuous steady state optimization using first optimization strategy.

	\mathbf{p}^*	\mathbf{p}_0		\mathbf{x}^*	\mathbf{x}_0
$F[1/h]$	0.0496	0.001	$X[g/L]$	0.7238	0.1924
$F_C[mol/L]$	22.2558	10	$C[mol/L]$	1.2530	0.0316
$F_N[mol/L]$	0.4170	1	$N[mol/L]$	0.0551	0.9038
$F_O[mol/L]$	8.9434	2	$O[mol/L]$	0.4895	0.0636
			$L[g/L]$	2.3852	0
			$E[mol/L]$	0.0283	1.8156
$P(\mathbf{p})[\$/(L \times h)]$	1.6351	-0.0884	$COX[mol/L]$	8.5105	5.5676

In this case, finite differences with a $\delta = 1 \times 10^{-6}$ take 81 seconds and 69 iterations to find suboptimal point $\mathbf{p}^* = [0.0484, 23.02, 0.422, 9.01]$ with an objective function value of $\$1.6344/(L \times h)$. Next, the second strategy of optimization is implemented and the results are presented in Table 8.11. In Table 8.11 $\mathbf{x} = [X, C, N, O, L, E, COX, F, F_C, F_N, F_O]$ and \mathbf{f} is the steady state constraint.

The second strategy of optimization presents more numerical problems. In the second and third start points, the method diverges. This is no surprise given that Solvopt takes constraints into account using an exact penalization method and then solving an unconstrained optimization problem [80]. If the penalty factor is not well chosen, the optimizer can encounter an unbounded problem and diverge. In the first point, the method is able to find a local minima but warns that the function is very steep. In the last one, the method is able to refine the optimal point found using the first strategy (with some rounding) to find a solution as good as that optimal point.

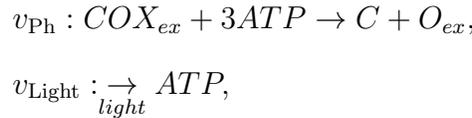
Table 8.11: Comparison of optimization results with initial point for continuous steady state optimization using second optimization strategy.

\mathbf{x}_i	[1, 1, 0.25, 1, 1, 1, 5, 0.5, 30, 0.5, 3]
\mathbf{x}^*	[0.0007, 0.968, 0.583, 0.862, 1.21, 1.112, 5, 0.001, 29.97, 0.479, 3.125]
$\max(\mathbf{f}(\mathbf{x}_i))$	13.1607
$\max(\mathbf{f}(\mathbf{x}^*))$	0.0005
$P(\mathbf{x}_i)_{[\frac{\$}{(L \times h)}]}$	-3.5
$P(\mathbf{x}^*)_{[\frac{\$}{(L \times h)}]}$	0.0034
Time [s]	0.7
Warnings	Function extremely steep at optimum.
\mathbf{x}_i	[5, 15, 0.3, 1, 0, 0, 0, 1, 10, 1, 3]
\mathbf{x}^*	NA
$\max(\mathbf{f}(\mathbf{x}_i))$	12.4751
$\max(\mathbf{f}(\mathbf{x}^*))$	NA
$P(\mathbf{x}_i)_{[\frac{\$}{(L \times h)}]}$	-29.5
$P(\mathbf{x}^*)_{[\frac{\$}{(L \times h)}]}$	NA
Time [s]	18
Warnings	Function is unbounded.
\mathbf{x}_i	1
\mathbf{x}^*	NA
$\max(\mathbf{f}(\mathbf{x}_i))$	1.3393
$\max(\mathbf{f}(\mathbf{x}^*))$	NA
$P(\mathbf{x}_i)_{[\frac{\$}{(L \times h)}]}$	-52.5
$P(\mathbf{x}^*)_{[\frac{\$}{(L \times h)}]}$	NA
Time [s]	18
Warnings	Function is unbounded.
\mathbf{x}_i	[0.724, 1.253, 0.055, 0.49, 2.385, 0.028, 8.51, 0.05, 22.256, 0.417, 8.943]
\mathbf{x}^*	[0.718, 1.239, 0.0552, 0.4949, 2.377, 0, 8.429, 0.0497, 22.047, 0.414, 8.924]
$\max(\mathbf{f}(\mathbf{x}_i))$	3.4×10^{-4}
$\max(\mathbf{f}(\mathbf{x}^*))$	8.9×10^{-9}
$P(\mathbf{x}_i)_{[\frac{\$}{(L \times h)}]}$	-1.6345
$P(\mathbf{x}^*)_{[\frac{\$}{(L \times h)}]}$	-1.6351
Time [s]	1.5
Warnings	Function is flat at optimum.

It is important to notice that when the steady state equations are added as constraints, the equation solving is performed by the optimization algorithm. If a good equation solving algorithm is available, the formulation used in the first optimization strategy may be more amenable. Given that the second strategy does not need to solve the dynamical system, the second strategy seems to be much faster.

8.1.4 Optimal Design of a Continuous Process Operating at Cyclic Steady State

The last class of problems considered in this chapter are those that attain a cyclic steady state. Some processes operate at cyclic steady state, for example, outdoors cultivation of photosynthetic organisms such as microalgae (see Chapters 4 and 5). Here, we shall modify the toy metabolic model by adding the following reactions,



where the upper bounds of these reactions are given by:

$$v_{\text{Ph}}^{UB}(\mathbf{x}) = \max\left(0, 0.1 \frac{COX}{1 + COX}\right),$$

$$v_{\text{Light}}^{UB}(t) = 0.5 \max\left(0, \sin\left(\frac{2\pi t}{24}\right)\right).$$

Considering that the first optimization strategy in Section 8.1.3 was more reliable, a very similar strategy is used in this example. The optimization variables are only the parameters F, F_C, F_N, F_O . Not all parameters lead to feasible solutions: some parameters lead to simulations where the physical variables X, C, N, O, L, E, COX attain cyclic steady state, but the penalty increases at each cycle. Therefore, cyclic steady state conditions were only applied to the physical variables whereas the penalty condition was added as a constraint. Another difference with the steady state case is that the objective function value varies with time. Therefore, instead of looking at the steady state value of the objective, the objective needs to be integrated through the 24 hour cycle. The optimization problem being solved looks like this:

$$\begin{aligned}
& \min - \int_0^{24} P(t, \mathbf{p}) dt \\
& \text{s.t. } P(t, \mathbf{p}) = F(\mathbf{p}) (10E_{css}(t, \mathbf{p}) + 50L_{css}(t, \mathbf{p}) - 2F_C(\mathbf{p}) - 5F_N(\mathbf{p}) - 0.5F_O(\mathbf{p})^2), \\
& [F(\mathbf{p}), F_C(\mathbf{p}), F_N(\mathbf{p}), F_O(\mathbf{p})] = \mathbf{p}, \\
& \alpha(\mathbf{p}) = 0,
\end{aligned}$$

where $\alpha(\mathbf{p})$ refers to the penalty function in (8.8) and the subscript *css* refers to cyclic steady state values.

Given a dynamical system with a known period T of the form,

$$\dot{\mathbf{y}}(\mathbf{y}(t, \mathbf{p}), \mathbf{p}) = \mathbf{f}(t, \mathbf{y}(t, \mathbf{p}), \mathbf{p}), \quad \forall t \in [0, t_f], \quad \mathbf{y}(0, \mathbf{p}) = \mathbf{p},$$

the system attains cyclic steady state if for any t [142]:

$$\mathbf{y}(T + t, \mathbf{p}) - \mathbf{y}(t, \mathbf{p}) = \mathbf{0}.$$

Given that in this example the period of the cyclic steady state is known and corresponds with the day 24 hour cycle, no phase-locking conditions like Eq. (2.1) in [142] are needed. The cyclic steady state conditions define an implicit function $\mathbf{x}(\mathbf{p}) = \mathbf{h}(\mathbf{p})$. Therefore, once again Theorem 8.1.1 is applicable. Let $\hat{\mathbf{x}} = [X \ C \ N \ O \ L \ E \ COX]^T$ and $\mathbf{x}_t(\mathbf{p}, \mathbf{x}_0) = \mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)$. The implicit function is given by $\mathbf{x}_T(\mathbf{p}, \mathbf{x}_0) - \mathbf{x}_0 = 0$ and given a fixed \mathbf{M} , Equation (8.9) is equivalent to:

$$\mathbf{x}'_T(\mathbf{p}, \mathbf{x}_0; \mathbf{M}, \mathbf{N}(T)) - \mathbf{N}(0) = \mathbf{0}, \quad (8.18)$$

which is equivalent to letting the sensitivities attain cyclic steady state. However, it has been observed that oscillating systems take a longer time to attain cyclic steady state than systems attaining steady state. Therefore, a hybrid approach was used: the dynamical system and its sensitivities were run for a considerable period of time (240 hours which is equal to 10 cycles), and then an equation solving approach us-

ing Lemma 3.5 in [124] was used to refine the solution. The nonsmooth system of equations was solved using the semismooth Newton method [103] and when the Jacobian became singular, iterations were taken using the LP-Newton method [33]. The results in these subsection were computed with relative and absolute integration tolerances of 1×10^{-7} and LP feasibility and optimality tolerances of 1×10^{-7} . Table 8.12 presents a summary of results. Once again, LD-derivatives present a significant advantage with respect to using finite differences in number of iterations and total time of optimization.

Table 8.12: Summary of results for the optimal design of a cyclic steady state system.

	LD-derivatives	FD ($\delta = 1E - 6$)
\mathbf{p}_0	[0.1, 10, 1, 1]	[0.1, 10, 1, 1]
\mathbf{p}^*	[0.0267, 157.4, 3.651, 11.06]	[0.0267, 157.5, 3.653, 11.07]
$\alpha(24, \mathbf{p}_0)$	0	0
$\alpha(24, \mathbf{p}^*)$	0	0
$P(\mathbf{p}_0)[\frac{\$}{(L \times day)}]$	-35.15	-35.15
$P(\mathbf{p}^*)[\frac{\$}{(L \times day)}]$	171.34	171.34
Time [s]	2900	5600
# Iterations	50	88
Warnings	Normal termination	Normal termination
\mathbf{p}_0	[5, 20, 1, 5]	
\mathbf{p}^*	[0.0254, 167.654, 3.867, 11.585]	
$\alpha(24, \mathbf{p}_0)$	0	
$\alpha(24, \mathbf{p}^*)$	0	
$P(\mathbf{p}_0)[\frac{\$}{(L \times day)}]$	-6900	
$P(\mathbf{p}^*)[\frac{\$}{(L \times day)}]$	186.83	
Time [s]	3956	
# Iterations	44	
Warnings	Normal termination	

8.1.5 Optimization of a continuous steady state process using genome-scale metabolic networks

In the following example, *E. coli* and yeast are grown together to produce ethanol from glucose and a mix of glucose and xylose. In this example, genome-scale metabolic networks are used: iJR904 [107] for *E. coli* (761 metabolites and 1075 reactions) and iND750 [31] for yeast (1061 metabolites and 1266 reactions). The optimization

variables are the dilution rate F , the feed composition and the O_2 mass transfer coefficient. The dynamic model is the following:

$$\begin{aligned} \dot{X}_i(t, \mathbf{p}, \mathbf{x}_0) &= \mu_i(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))X_i(t, \mathbf{p}, \mathbf{x}_0) - F(\mathbf{p})X_i(t, \mathbf{p}, \mathbf{x}_0), \\ \dot{S}_j(t, \mathbf{p}, \mathbf{x}_0) &= - \sum_i \mathbf{v}_{i,j}(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0))X_i(t, \mathbf{p}, \mathbf{x}_0) + F(S_{j,0} - (\mathbf{p})S_j(t, \mathbf{p}, \mathbf{x}_0)) \\ &\quad + MT_j(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)), \\ \mathbf{S}_0 &= \mathbf{f}(\mathbf{p}), \mathbf{x}(0, \mathbf{p}, \mathbf{x}_0) = \mathbf{x}_0, \text{ for } i = E. coli, \text{ yeast, for } j = G, X, O, E, \end{aligned} \tag{8.19}$$

where $\mathbf{x} = [X_B, X_Y, G, X, O, E, \alpha]$ (bacterial biomass, fungal biomass, glucose, xylose, oxygen, ethanol, and penalty, respectively), and $MT_j = 0$ for all j except O where $MT_O(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)) = p_4(0.21K_{H,O_2} - O(\mathbf{x}(t, \mathbf{p}, \mathbf{x}_0)))$, $\mathbf{S}_E = 0$, $\mathbf{S}_G = p_2 + p_3$, $\mathbf{S}_X = p_2/2$ and $F = p_1$. p_2 is the amount of glucose in a glucose/xylose mix concentration containing two thirds glucose and one third xylose by weight, p_3 is the concentration of a pure glucose stream, p_4 is the O_2 mass transfer coefficient, and p_1 is the dilution rate in units of h^{-1} .

In this problem we assumed the following prices:

1. Ethanol: assumed price of \$0.391/kg of diluted ethanol. Pure ethanol has a price of \$0.813/kg [62].
2. 66% glucose/33% Xylose mix: \$0.07/kg [113].
3. Pure glucose: \$0.126/kg [5].

The objective function is given by:

$$P(\mathbf{p}) = p_1(0.391E_{ss}(\mathbf{p}) - 0.07 \times 1.5p_2 - 0.126p_3)$$

First, this problem was solved using the first strategy described in subsection 8.1.3 using solvers Solvopt [80] and Ipopt [139]. Ipopt is designed to solve twice continuously differentiable systems, which cannot be satisfied by nonsmooth systems. Here, we provide an element of the B -subdifferential as derivative information. How-

ever, dual feasibilities which enforce optimality conditions have to be relaxed because nonsmooth systems are unable to satisfy them.

The results can be found in Table 8.13. In this case, Solvopt seems to perform better than Ipopt. This can be explained by the fact that whereas Solvopt was designed for nonsmooth optimization, Ipopt was not. Therefore, Ipopt cannot verify optimality conditions (enforced as dual feasibilities). Given the nonsmooth nature of the problem, a tolerance of 0.1 for dual feasibility was used. Therefore, early termination is very probable. However, the tolerance cannot be tightened much as nonsmooth systems are unable to satisfy dual feasibility at a nonsmooth point.

Table 8.13: Summary of results for the optimal design of a steady state system producing ethanol with *E. coli* and yeast.

Solvopt				
Iter.	\mathbf{p}_0	$P(\mathbf{p}_0)[\$/h]$	\mathbf{p}^*	$P(\mathbf{p}^*)[\$/h]$
49	[0.05,10,0,0.001]	1.63×10^{-5}	[0.0926,30,0,0.0031]	7.52×10^{-5}
65	[1,30,30,0.1]	-6.93×10^{-3}	[0.0926,30,0,0.0031]	7.52×10^{-5}
Ipopt				
Iter.	\mathbf{p}_0	$P(\mathbf{p}_0)[\$/h]$	\mathbf{p}^*	$P(\mathbf{p}^*)[\$/h]$
16	[0.05,10,0,0.001]	1.63×10^{-5}	[0.143,11.03,0.0005,0.003]	3.52×10^{-7}
23	[1,30,30,0.1]	-6.93×10^{-3}	[0.1056,22.65,4.51,0.003]	4.47×10^{-5}

8.2 Conclusions

This chapter integrates the theory in Chapters 6 and 7 to solve optimization problems. This chapter illustrates different classes of DFBA problems, such as parameter estimation problems and optimal design of batch and continuous processes. Different strategies to solve these problems efficiently have been presented. Throughout the examples in this chapter, LD-derivatives present a better performance than finite differences, justifying the work presented in Chapters 6 and 7. LD-derivatives work well with nonsmooth optimization code Solvopt [80]. The last example illustrates the optimization of a DFBA problem including two genome-scale metabolic networks, which illustrates the power of the modeling framework presented throughout this thesis.

DFBA optimization problems are nonconvex. In the future, global optimization

strategies that are able to solve these kinds of problems will be developed. In this thesis we take two critical steps to enable global optimization:

1. This thesis presents a way of performing local optimization reliably. Local optimization is needed as part of global optimization strategies.
2. Appendix C contains the derivation of convex and concave relaxations for lexicographic LPs.

Nevertheless, a lot can be gained just by performing rigorous local optimization as has been shown in the examples of this chapter.

Chapter 9

Conclusions and Future Work

Due to the many applications of microbial communities in industrial bioprocesses (e.g. the food, pharmaceuticals, and biofuels industries) as well as their appearance in natural ecosystems (e.g. oceans or the gut microbiome), the modeling and optimization of bioprocesses employing microbial communities is critical. The modeling, optimization and control of these bioprocesses is a challenging task because microorganisms are very complex systems by themselves and bioprocesses span several time and length scales. In particular, microorganisms often operate at much faster time-scales and shorter length-scales than the macrosystem in which the bioprocesses or the ecological processes take place. Therefore, accurate and reliable mathematical models are difficult to obtain, and good multi-scale models are often nonsmooth.

Most bioprocess models used today in chemical engineering settings remain rather simplistic and based on unstructured models. These models rely heavily on experimental data and have limited application due to the many metabolic states microorganisms attain in these processes. As such, unstructured models are inaccurate predictors of growth for most realistic situations, such as those involving multiple nutrient limitations, day/night transitions, and competitive and symbiotic relationships. Meanwhile, high-throughput genome sequencing techniques have led to the development of genome-scale metabolic network reconstructions (GENREs) for several microorganisms. The information in these networks can be incorporated into process models using flux balance analysis (FBA) [137, 98] and dynamic flux balance

analysis (DFBA) [137, 87].

DFBA models address the limitations of unstructured models by considering all possible metabolic states included in a GENRE. Given the extracellular environment, FBA determines for each microorganism the metabolic state or combination of metabolic states that attain maximum growth. GENREs also consider all minor nutrients in a network and can be used to model symbiotic and competitive relationships between microorganisms [77]. As such, DFBA can be used to construct reliable process models, find an optimal design, and control complex bioprocesses.

DFBA models result in dynamical systems with linear programs (LPs) embedded [56, 59]. These systems are difficult to simulate due to LPs having nonunique solutions and becoming infeasible under environmental conditions that do not support growth. In addition, these systems are difficult to optimize due to their nonsmooth nature which makes obtaining reliable sensitivity information challenging. The work described in this thesis is a critical contribution in the field of bioprocess modeling as it enables the simulation and optimization of DFBA bioprocess models. Next, the main contributions of this thesis are described.

In Chapter 3, a Matlab-based DFBA simulator is introduced. This simulator combines the ideas of lexicographic optimization described in [56, 59] with the Phase I of the simplex method [12] to result in a more robust simulator with a penalty function useful for optimization purposes. This simulator currently has more than a 100 academic users, and it has resulted in a book chapter [7] (Appendix A) and a workshop. In addition, this simulator has triggered collaborations with Prof. Michael Henson of UMass Amherst and Prof. Ahmed Al Hajaj of Masdar Institute in the United Arab Emirates (UAE) that have resulted in the implementation of DFBA models in different settings such as the modeling of chronic wound biofilms, syngas fermentation in bubble column bioreactors [22, 23], and algae cultivation in the harsh environment of the UAE. As a result of this doctoral project, complex DFBA models are now used in more varied settings by researchers with a strong biological background but not as strong numerical background, allowing them to benefit from the power of mathematical modeling as a tool for better process control and design.

In Chapters 4 and 5, DFBA_{lab} is used to explore new algal biomass cultivation strategies. In particular, these chapters show that algae cultivation as a CO₂ capture strategy is inefficient, the growth of algae from flue gas is economical only at very short distances from the flue gas source, and that microbial consortia design has great potential for biomass cultivation. In particular, algae can be grown with a heterotroph creating a symbiotic relationship that boosts biomass productivity and that provides a biological route for transforming lignocellulosic waste into biolipids. With microbial consortia, algae cultivation can become economical in many more places and not only in the proximity of power plants. In addition, the carbon balances in these chapters illustrate where carbon is being lost in the system and how the productivity of the system can be improved. Despite the lack of experimental validation, the results in these chapters provide a future direction in algae cultivation experiments. This direction will be pursued by the group of Prof. Ahmed Al Hajaj in the UAE.

With better DFBA simulation tools, the optimization of DFBA models becomes possible. Optimization problems with DFBA models embedded can be formulated in the following manner:

$$\begin{aligned}
\min_{\mathbf{p}} J(\mathbf{p}) &\equiv \theta(\mathbf{x}_{t_f}(\mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} \varphi(t, \mathbf{x}_t(\mathbf{p}), \mathbf{p}) dt & (9.1) \\
\text{s.t. } \mathbf{G}(\mathbf{p}) &\equiv \mathbf{g}(\mathbf{x}_{t_f}(\mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} \mathbf{h}(t, \mathbf{x}_t(\mathbf{p}), \mathbf{p}) dt \leq \mathbf{0}, \\
\mathbf{p} &\in S \subset \mathbb{R}^{n_p},
\end{aligned}$$

where $\mathbf{x}_t(\mathbf{p}) \equiv \mathbf{x}(t, \mathbf{p})$. This optimization problem can be nonconvex and nonsmooth. To find a global minimum, global optimization strategies are needed. Local optimization is a prerequisite for robust global optimization strategies to be formulated. The work in this thesis has been focused on local optimization of DFBA systems. Appendix C presents convex and concave relaxations for lexicographic LPs, which are also needed for global optimization.

The embedded LPs introduce two challenges to this optimization problem: nonsmoothness and implicit constraints. The solution to DFBA models can be nonsmooth,

and therefore, the objective function value and the constraints in optimization problem (9.1) can be nonsmooth. Nonsmooth optimization algorithms, such as bundle methods [89] and modified versions of Shor’s r-algorithm with space dilation [80] exist among other nonsmooth optimization methods, but they require generalized derivative information. Chapters 6 and 7 present the work required to obtain reliable sensitivity information of DFBA models.

In this thesis we have used the notion of lexicographic differentiation [96] and lexicographic directional derivatives (LD-derivatives) [72, 74] to compute generalized derivative information. LD-derivatives possess desirable properties, such as satisfying a sharp chain rule, and being as useful as Clarke’s generalized Jacobian for optimization and equation solving purposes. The computation of LD-derivatives for the objective function value of a LP as a function of its right-hand side has been presented in [60], but its extension to LLPs is not obvious. In particular, computing LD-derivatives of the objective function values of a LLP as a function of its right-hand side involves computing directional derivatives at the boundaries of closed sets, a case that is not handled by classical theory. In Chapter 6 the mathematical derivation of LD-derivatives of LLP objective function values as a function of its right-hand side is presented. These LD-derivatives have been used to compute the steady-state of a bubble column bioreactor, optimize the behavior of a supplier selling products to two companies, and solve for the steady-state of a continuous process involving *E. coli* cultivation. In addition, this chapter presents conditions under which the LD-derivatives of $[\mathbf{f} \circ \mathbf{g}]$ at \mathbf{x} can be computed even if $\mathbf{g}(\mathbf{x})$ is at the boundary of the domain of \mathbf{f} .

The computation of LD-derivatives for ODE systems with nonsmooth right-hand sides, such as DFBA models, has been presented in [72]. The method relies on integrating the LD-derivatives of the nonsmooth right-hand side of the ODE system. The LD-derivatives of DFBA models can be obtained as solutions of related ODE systems. The work in [73] enables the formulation of a numerical method to compute the sensitivities of ODE systems with abs-factorable right-hand sides [71], but it is not readily applicable to DFBA models. The work in Chapter 7 bridges the gap between

the methods presented in [71] and the ones required for DFBA models. Also, Chapter 7 provides alternative methods to compute the LD-derivatives of LLPs, which can be used when the ones presented in Chapter 6 present numerical difficulties.

Another difficulty introduced to (9.1) by the LP is the nature of the feasible set S , which contains implicit constraints regarding the feasibility of the embedded LP. Obtaining an explicit representation of S is very challenging, if possible. Instead, the strategies in this thesis use the penalty function presented in Chapter 3 to incorporate these implicit constraints into (9.1).

Chapter 8 presents different optimization problems with DFBA models embedded:

1. Parameter estimation problems;
2. Optimal batch system design;
3. Optimal steady-state system design;
4. Optimal cyclic steady-state system design.

Chapter 8 uses the work in Chapters 6 and 7 as well as the theory in [75] and [124] to optimize DFBA models. Optimization strategies for the different classes of problems are illustrated using a toy metabolic network. In addition, an optimization problem using GENREs is presented. The nonsmooth optimizer Solvopt [80] and the interior-point method Ipopt [139] are used to solve these optimization problems.

With this thesis, the vision described in Figure 9-1 comes closer to reality. In this vision, DFBA becomes an accessible tool for future bioprocess engineers for better bioprocess design. In this way, the power of mathematical modeling can be used by bioprocess engineers to drive their experimental work and arrive at better bioprocesses in a shorter time frame and with less resources required.

Future work remains after this thesis. In particular, numerical challenges persist in the computation of sensitivities in DFBA systems. In addition, very few nonsmooth optimization solvers have been tested. A comparison in performance of different bundle solvers and nonsmooth optimizers remains to be done. In addition, the sensitivity

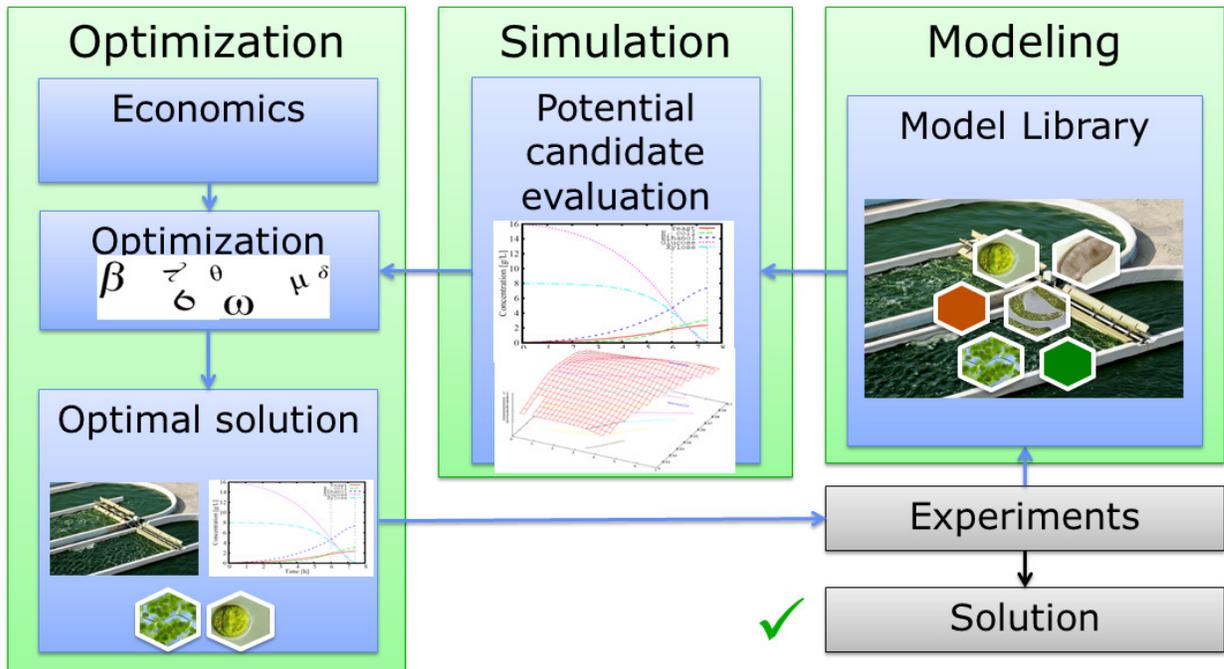


Figure 9-1: Vision for the future of bioprocess design. In the future, the GENREs for more species will be available and these metabolic models will increase in level of detail. In addition, better bioprocess models for different settings (bioreactors, raceway ponds, oceans) will be developed. Together, a library of process models and GENREs will be put together. The bioprocess engineer will be able to test novel combinations of microorganisms and process models and evaluate their potential before performing any bench-scale test. With the help of mathematical optimization, an optimized process design can be obtained. This design can then be tested at the bench and pilot scales, and the experimental results can be compared with the DFBA predictions. If experimental and simulation results are very different, the library model can be refined using data analytics and machine learning and the optimization process can be repeated. In this way, DFBA can guide the experimental design of the bioprocess engineer, the models can learn from the experimental data, and together DFBA and experiments can drive the design of better bioprocesses.

computation of DFBA models remains to be integrated into DFBAlab. This sensitivity information can be included using automatic differentiation, but this strategy can slow the performance of Matlab drastically. Finally, the model validation of the raceway pond presented in Chapters 4 and 5 remains to be done. Nevertheless, the work in this thesis has brought us closer to making the vision described in Figure 9-1 a reality in the near future.

Appendix A

Dynamic Flux Balance Analysis using DFBAlab

This Appendix is a reproduction of [7].

Dynamic flux balance analysis (DFBA) [137, 87] is a bioprocess modeling framework that relies on genome-scale metabolic network reconstructions (GENREs) of microorganisms. It is the dynamic extension of flux balance analysis (FBA) [98], which has become popular with the advent of high-throughput genome sequencing. In fact, the number and level of detail of genome scale metabolic network reconstructions has rapidly increased since 1999 (see Fig.1 in [93]). Despite the ever-increasing availability of new and better metabolic network reconstructions, DFBA modeling remains challenging, and therefore its use has been limited.

Traditionally, bioprocess modeling relies on unstructured models to calculate the growth rates of microorganisms. This approach has significant limitations that make it impossible to simulate very complex bioprocesses. These limitations are countered by FBA by considering genome-scale metabolic networks of the microorganisms involved. FBA models the growth and metabolic fluxes rates of microorganisms as solutions of the following linear program (LP):

$$\begin{aligned}
& \max v_{growth} \\
& \text{s.t. } \mathbf{S}\mathbf{v} = \mathbf{0}, \\
& \mathbf{v}^{LB} \leq \mathbf{v} \leq \mathbf{v}^{UB},
\end{aligned}$$

where \mathbf{S} is the stoichiometric matrix, \mathbf{v} is the fluxes vector, \mathbf{v}^{UB} and \mathbf{v}^{LB} are the bounds on the metabolic fluxes given by thermodynamics, the extracellular environment and/or genetic modifications, and v_{growth} sums all growth associated fluxes. Given mass balance and thermodynamic constraints on a metabolic network, FBA finds a solution that satisfies these constraints and maximizes growth. If the LP becomes infeasible, it may indicate a lack of sufficient substrates and nutrients to provide the minimum maintenance energy for the respective microorganism to survive.

DFBA combines process models described by an ordinary differential equation (ODE) system, a differential-algebraic equation (DAE) system, or a partial differential-algebraic equation (PDAE) [22, 23] system with FBA to model bioprocesses. These models can be expressed as dynamic systems with LPs embedded [59, 56] which are challenging to simulate. The embedded LP poses difficulties in the form of non-unique solutions and premature LP infeasibilities. Fortunately, these complications have been addressed by efficient DFBA simulators, DSL48LPR in FORTRAN [59] and DFBAlab in MATLAB [44]. Both are free for academic research and can be found in the following webpage: <http://yoric.mit.edu/software>. The rest of this chapter will talk exclusively about how to use DFBAlab to perform DFBA simulations.

DFBAlab uses lexicographic optimization and the phase I of the simplex algorithm to deal with nonunique solutions and LP infeasibilities, respectively. Lexicographic optimization is a strategy that enables obtaining unique exchange fluxes. This strategy requires defining an objective function for each exchange flux of interest. More information can be found in [44] and [59].

A.1 Materials

1. MATLAB: DFBAlab is compatible with MATLAB 7.12.0 and newer versions.
2. DFBAlab: DFBAlab is a MATLAB code that performs DFBA simulations efficiently and reliably. It is free for academic research and can be downloaded at <http://yoric.mit.edu/software>. DFBAlab contains three folders named “Functions”, “Examples_Direct”, and “Examples_DAE”. The folder named “Functions” must be added to the MATLAB path. The Examples folders contain the examples described in [44]. It is recommended to use “Examples_DAE” since these examples run much faster than “Examples_Direct”.
3. Relevant GENREs: To perform DFBA simulations, GENREs of the relevant microorganisms are needed. An extensive collection of GENREs can be found at the following webpage: <http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms>. There are some protocols and methods that generate these reconstructions automatically. A list can be found at Table 1 of [40]. GENREs are usually published in the Systems Biology Markup Language (SBML) [61].
4. LP Solvers Gurobi or CPLEX: As mentioned before, DFBA calculates the growth rates and exchange fluxes rates of microorganisms by solving LPs. Therefore, LP solvers are needed. DFBAlab is currently compatible with CPLEX [28] and Gurobi [50]. Academic licenses are available for both LP solvers.
5. Cobra Toolbox: The Cobra Toolbox [118] is a package that runs in MATLAB that uses constraint-based modeling to predict the behavior of microorganisms. It can be obtained from <https://opencobra.github.io/>. The Cobra toolbox has a function that transforms a metabolic network in SBML format into a “.mat” format, which is the input used by DFBAlab. For this function to work, the SBML package must be obtained from http://sbml.org/Main_Page.

A.2 Methods

Here, we describe step by step how to model a DFBA problem using DFBAlab. The examples in “Examples_Direct” and “Examples_DAE” require the same modeling effort. Only the implementation in “Examples_DAE” will be discussed because the implementation in “Examples_Direct” is very similar. Examples in both folders contain three key files: main.m, DRHS.m, and RHS.m. In addition, the examples in “Examples_DAE” contain a file called evts.m. In addition, the Examples folders contain a folder called “Example-ModelImport”. This folder contains a very simple script that converts an SBML file into a “.mat” file that can be used in MATLAB simulations. Next, we show the relevant parts of these files and the inputs required from the user. In addition, we use Example 3 in [44] to illustrate the use of DFBAlab. In this example, *Chlamydomonas reinhardtii* and *Saccharomyces cerevisiae* grow together in a pond open to the atmosphere. For simplicity, this pond is modeled as a single continuously stirred-tank reactor (CSTR). The model results in a DAE system as pH balances are considered. In Section A.3, some common problems regarding the use of DFBAlab are described.

A.2.1 Converting a GENRE in SBML format into “.mat” format

When published, GENREs are usually in SBML format. The file “ModelImport.m” in the “Example-ModelImport” folder in Examples calls the Cobra toolbox and converts GENREs in SBML format into “.mat” format. This function is really easy to use. Just modify line 4 of the code,

```
NAME1=readCBModel('NAME2',DB,'SBML'),
```

and replace NAME1 for the name you want to give your model, replace NAME2 with the name of the SBML file, and replace DB with the numerical value you want infinity to be replaced with. Usually, a value of $DB = 1000$ works fine. The output will be a model with the name NAME1.mat. Every microorganism that needs to be modeled in DFBAlab must be transformed into a “.mat” file.

A.2.2 Inputs for the main.m file

The main.m file sets up the simulation. In this section, each part of the main.m file will be described and the required inputs will be specified. It is important to notice that the structure variable INFO contains important information that is then passed to other MATLAB files and functions. This structure can be used to pass other important parameter information to DRHS.m, RHS.m, and evts.m. The first part of main.m clears the workspace and specifies the number of models that will be used in the simulation.

```
clear all
```

```
INFO.nmodel = NM;
```

Here, NM stands for the number of models in the simulation. For example, consider a dynamic model of a plug flow reactor (PFR) with two different species growing. The PFR can be discretized in N spatial slices to transform it from a partial differential equation (PDE) system to an ODE system. Then, $NM = 2*N$. The next part of the main.m file loads the models. Here, the models must already be in “.mat” format after going through the “ModelImport.m” code:

```
load NAMEMODEL.mat
```

```
model{i} = NAMEMODEL;
```

```
DB(i) = db;
```

```
INFO.DB = DB;
```

Each species must be loaded into the file using the load command and replacing NAMEMODEL for the actual name of the model. Then, for $i = 1, \dots, nmodel$, the cell model stores each relevant metabolic network model. The array DB stores the default bound specified in ModelImport.m for each model. In this way, DFBAlab can change this bound information back to infinity. This array is stored in the INFO structure. The next part defines the exID cell, which carries the exchange fluxes information:

```
exID{i}=[ex1,ex2,,exj];
```

```
INFO.exID = exID;
```

Again $i = 1, \dots, nmodel$, and $ex1, ex2, exj$ correspond to the indices of the exchange

fluxes that are needed in the DRHS.m file for model i . For example, let us assume that model 1 corresponds to an *E. coli* model. To model this bioprocess, we may care about the exchange fluxes of glucose, oxygen, and carbon dioxide. Then in `model{1}.rxns` the indices associated with the exchange fluxes for glucose, oxygen, and carbon dioxide must be found. These are the indices that must be inserted in place of $ex1, ex2, exj$. The order of these indices is relevant for the RHS.m file as explained later.

Next, the cell C corresponding to the cost vectors is defined:

```

minim = 1;
maxim = -1;
% Maximize growth
Ci(j).sense = maxim;
Ci(j).rxns = [cin];
Ci(j).wts = [1];

```

DFBALab relies on lexicographic optimization to perform efficient DFBA simulations [44]. Each cost vector requires three entries: sense, reactions, and weights. The information in $C\{i\}(j)$ corresponds to model i and cost vector j . Usually, the first cost vector for any model will correspond to maximization of biomass. Other relevant biological objectives can be listed as subsequent cost vectors. All the fluxes that appear in the right-hand side of the ODE system must be listed as cost vectors in order to guarantee a unique solution of the simulation. If you want the corresponding cost vector to be maximized, enter `maxim` in `sense`, otherwise enter `minim`. Then in $C\{i\}(j).rxns$ list the indices of the reactions corresponding to this cost vector. Finally in $C\{i\}(j).wts$ enter the coefficients corresponding to this cost vector. For example,

```

C{i}(j).sense = maxim;
C{i}(j).rxns = [1108,103];
C{i}(j).wts = [1,-1];
INFO.C = C;

```

is equivalent to defining cost vector j for model i as maximize $v_{1108} - v_{103}$. The cost vectors are stored in the INFO structure. Next the initial conditions, integration time

```

and DFBALab options are set. Y0 = [ICVECTOR];
% Time of simulation
tspan = [ti,tf];
% LP Objects construction parameters
INFO.LPsolver = 0; % CPLEX = 0, Gurobi = 1.
INFO.tol = tol1; % Feasibility, optimality and convergence tolerance
for Cplex (tol>=1E-9).
% It is recommended it is at least 2 orders of magnitude
% tighter than the integrator tolerance.
% If problems with infeasibility messages, tighten this
% tolerance.
INFO.tolPh1 = tol2; % Tolerance to determine if a solution to phaseI
equals zero.
% It is recommended to be the same as INFO.tol.
INFO.tolevt = tol3; % Tolerance for event detection. Has to be greater
% than INFO.tol. Recommended to be 2 times the integration tolerance.

```

$Y0$ is a column vector containing the initial conditions. $tspan$ contains the integration time interval: ti corresponds to the initial time and tf corresponds to the final time. Next, some DFBALab parameters are established: $INFO.LPsolver$ selects between Gurobi (enter 1) and CPLEX (enter 0). Next, the LP tolerance is set at $INFO.tol$. This tolerance has to be larger than 10^{-9} . Next, $INFO.tolPh1$ corresponds to the threshold value under which the penalty function will be considered equal to zero. Here, we recommend using $INFO.tol$ for this value. Finally, $INFO.tolevt$ corresponds to the event detection tolerance that triggers when the FBA LP must be solved again. Here, we recommend using two times the absolute tolerance of the integration method. Next, the integration options are set:

```

M = [MASS];
options = odeset('AbsTol',tol4,'RelTol',tol5,'Mass',M,'Events',@evts);

```

If we are integrating a DAE system, a mass matrix must be defined. Using `odeset`, other MATLAB integration options can be set such as absolute and relative toler-

ances, nonnegativity constraints, and event detection. DFBAlab always uses event detection if you use the Examples in “Examples_DAE”. The following parts of the “main.m” file do not require any further inputs.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[model,INFO] = ModelSetupM(model,Y0,INFO);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if INFO.LPsolver == 0
[INFO] = LexicographicOpt(model,INFO);
elseif INFO.LPsolver == 1
[INFO] = LexicographicOptG(model,INFO);
else
display('Solver not currently supported.');
```

```

end
tic
tint = 0;
TF = [];
YF = [];
while tint<tspan(2)
% Look at MATLAB documentation if you want to change solver.
% ode15s is more or less accurate for stiff problems.
[T,Y] = ode15s(@DRHS,tspan,Y0,options,INFO);
TF = [TF;T];
YF = [YF;Y];
tint = T(end);
tspan = [tint,tspan(2)];
Y0 = Y(end,:);
if tint == tspan(2)
break;
end
```

```

% Update b vector
[INFO] = bupdate(tint,Y0,INFO);
%Determine model with basis change
value = evts(tint,Y0,INFO);
ind = find(value<=0);
fprintf('Basis change at time%d. ',tint);
k = 0;
ct = 0;
while isempty(ind)
k = k + 1;
ct = ct + size(model{k}.A,1);
ind2 = find(ind<=ct);
if isempty(ind2)
INFO.flagbasis = k;
fprintf('Model %i. \ n',k);
% Perform lexicographic optimization
if INFO.LPsolver == 0
[INFO] = LexicographicOpt(model,INFO);
elseif INFO.LPsolver == 1
[INFO] = LexicographicOptG(model,INFO);
else
display('Solver not currently supported. ');
end
ind(ind2)=[];
end
end
end
display(toc);

```

The function ModelSetupM takes the model cell and the INFO structure to trans-

form the LPs into the standard form, which is the key for efficient integration with DFBAlab. The functions `LexicographicOpt` and `LexicographicOptG` solve the LPs with CPLEX and Gurobi, respectively. Then comes the integration loop using the numerical integrator `ode15s`, used for stiff systems which is the case for DFBA systems. The results of integration are stored in the vectors `YF` for the states and `TF` for the times. After this, any plots can be generated using the information in `YF` and `TF`.

A.2.3 Sample inputs for `main.m`

The following MATLAB code is commented in the relevant sections.

```
clear all

INFO.nmodel = 2; % Number of models: one for algae and one for yeast.
load iND750.mat % This is the yeast model
model{1} = iND750;
DB(1) = 1000;% This is the default bound for the yeast mode
l (1000 = Infinity)
load iRC1080.mat % This is the algae model
model{2} = iRC1080;
DB(2) = 1000; % This is the default bound for the algae model
(1000 = Infinity)
INFO.DB = DB;
exID{1}=[428,458,407,420]; % These are the exchange fluxes with variable
bounds for the yeast model. The bounds are defined in RHS.m.
exID{2}=[26,28,24,25,1,2,3,4,5,6,7,8,9,10,11,12,13,62,63,64,65,27,81,47];
% These are the exchange fluxes with variable bounds for the algae model.
The bounds are defined in RHS.m.
INFO.exID = exID;% Next, we define the cost vectors for yeast and algae.
For both cases, first, we maximize growth, and then we maximize or
minimize each one of the exchange fluxes that appear in the ODE system.
minim = 1;
```

```

maxim = -1;
% Yeast
% Maximize growth
C{1}(1).sense = maxim;
C{1}(1).rxns = [1266];
C{1}(1).wts = [1];
% Glucose
C{1}(2).sense = maxim;
C{1}(2).rxns = [428];
C{1}(2).wts = [1];
% O2
C{1}(3).sense = maxim;
C{1}(3).rxns = [458];
C{1}(3).wts = [1];
% CO2 C{1}(4).sense = maxim;
C{1}(4).rxns = [407];
C{1}(4).wts = [1];
% Ethanol C{1}(5).sense = maxim;
C{1}(5).rxns = [420];
C{1}(5).wts = [1];
% Algae
% Maximize growth
C{2}(1).sense = maxim;
C{2}(1).rxns = [63];
C{2}(1).wts = [1];
% Acetate
C{2}(2).sense = minim;
C{2}(2).rxns = [28];
C{2}(2).wts = [1];
% O2

```

```

C{2}(3).sense = maxim;
C{2}(3).rxns = [24,81];
C{2}(3).wts = [1,1];
% CO2
C{2}(4).sense = minim;
C{2}(4).rxns = [25];
C{2}(4).wts = [1];

INFO.C = C;
% Initial conditions
% Y1 = Volume (L)
% Y2 = Biomass Yeast (gDW/L)
% Y3 = Biomass Algae (gDW/L)
% Y4 = Glucose (mmol/L)
% Y5 = O2 (mmol/L)
% Y6 = Total Carbon (mmol/L)
% Y7 = Ethanol (mmol/L)
% Y8 = Acetate (mmol/L)
% Y9 = Total Nitrogen (mmol/L)
% Y10 = Penalty
% Y11 = NH4+
% Y12 = NH3
% Y13 = CO2
% Y14 = HCO3-
% Y15 = CO3 -2
% Y16 = H +
Y0 = [140 1.1048 1.8774 0.0140, 0.00065156 1.2211 8.2068 0.0237
0.1643 0 0.1643 2.4476E-5 1.0568 0.1643 2.5842E-6 2.6701E-6]';
% Time of simulation
tspan = [0,24];

```

```

% LP options and tolerances
INFO.LPsolver = 0; % CPLEX = 0, Gurobi = 1.
INFO.tol = 1E-9; % Feasibility, optimality and convergence tolerance
for LP solver (tol>=1E-9). It is recommended it is at least 1 order of
magnitude tighter than the integration tolerance.
INFO.tolPh1 = INFO.tol; % Tolerance to determine if a solution to
phaseI equals zero. Usually, the best value here is to set it
equal to INFO.tol.
INFO.tolevt = 2E-6; % Tolerance for event detection. Has to be
greater than INFO.tol. Usually a value of two times the
integration tolerance works fine.
% This is a DAE system because of the pH balances. The first 10 states
are the differential states and the last 6 correspond to the
algebraic states.
M = [eye(10) zeros(10,6); zeros(6,16)];
options = odeset('AbsTol',1E-6,'RelTol',1E-6,'Mass',M,'Events',@evts);
% This part of the code constructs the LP problem structures that will
be solved during integration, and solves the LPs at the initial
conditions.
[model,INFO] = ModelSetupM(model,Y0,INFO); if INFO.LPsolver == 0
[INFO] = LexicographicOpt(model,INFO);
elseif INFO.LPsolver == 1
[INFO] = LexicographicOptG(model,INFO);
else
display('Solver not currently supported.');
```

end % This is the integration loop. tint = 0;

% TF and YF will concatenate T and Y that are returned by the DAE
numerical integrator. TF = [];

YF = [];

while tint<tspan(2)

```

% Look at MATLAB documentation if you want to change solver.
% DFBA systems tend to be stiff systems and ode15s is more or less
accurate for stiff problems.
[T,Y] = ode15s(@DRHS,tspan,Y0,options,INFO);
TF = [TF;T];
YF = [YF;Y];
tint = T(end);
tspan = [tint,tspan(2)];
Y0 = Y(end,:);
if tint == tspan(2)
break;
end

% Update the right-hand sides of the LPs given the current time and
states
[INFO] = bupdate(tint,Y0,INFO);
%Determine which LPs had a basis change
value = evts(tint,Y0,INFO);
ind = find(value<=0);
fprintf('Basis change at time%d. ',tint);
k = 0;
ct = 0;
while isempty(ind)
k = k + 1;
ct = ct + size(model{k}.A,1);
ind2 = find(ind<=ct);
if isempty(ind2)
INFO.flagbasis = k;
fprintf('Model %i. \ n',k);
% Perform lexicographic optimization

```

```

if INFO.LPsolver == 0
[INFO] = LexicographicOpt(model,INFO);
elseif INFO.LPsolver == 1
[INFO] = LexicographicOptG(model,INFO);
else
display('Solver not currently supported.');
```

end

```
ind(ind2)=[];
end
end
end
end
% TF contains all the times and YF all the states at these times.
Any plotting options can be included here.
```

A.2.4 Inputs for the DRHS.m file

The DRHS function defined in DRHS.m takes time, states and the INFO structure and returns the right-hand side vector of the ODE or DAE system:

```
function dy = DRHS(t, y, INFO)
```

This file is very flexible. A key command that takes place before the right-hand side values are set is:

```
[flux,penalty] = solveModel(t,y,INFO);
```

Here, the flux variable is a matrix with rows corresponding to each model and columns corresponding to each cost vector. Therefore, $flux(i, j)$ corresponds to the optimal value of cost vector j and model i with the order defined in main.m. The penalty vector contains the objective function values of the Phase I LPs. If $penalty(i) > 0$, model i corresponds to an infeasible LP. Otherwise, model i is feasible. We recommend that a penalty state that integrates the sum of all penalty functions is set. If this penalty state is greater than zero at the end of the simulation, then, this DFBA simulation is infeasible.

A.2.5 Sample inputs for the DRHS.m file

The following MATLAB code is commented in the relevant sections.

```
function dy = DRHS(t, y, INFO)
```

```
% Y1 = Volume (L)
```

```
% Y2 = Biomass Yeast (gDW/L)
```

```
% Y3 = Biomass Algae (gDW/L)
```

```
% Y4 = Glucose (mmol/L)
```

```
% Y5 = O2 (mmol/L)
```

```
% Y6 = Total Carbon (mmol/L)
```

```
% Y7 = Ethanol (mmol/L)
```

```
% Y8 = Acetate (mmol/L)
```

```
% Y9 = Total Nitrogen (mmol/L)
```

```
% Y10 = Penalty
```

```
% Y11 = NH4+
```

```
% Y12 = NH3
```

```
% Y13 = CO2
```

```
% Y14 = HCO3-
```

```
% Y15 = CO3 -2
```

```
% Y16 = H +
```

```
% Assign values from states
```

```
Vol = y(1);
```

```
X(1) = y(2);
```

```
X(2) = y(3);
```

```
for i=1:13
```

```
S(i) = y(3+i);
```

```
end % Feed rates Fin = 1;
```

```
Fout = 1;
```

```

% Biomass Feed concentrations
Xfeed(1) = 0;
Xfeed(2) = 0;

% Mass transfer coefficients
KhO2 = 0.0013;
KhCO2 = 0.035;

% Mass transfer expressions
MT(1) = 0;
MT(2) = 0.6*(KhO2*0.21*1000 - S(2));
MT(3) = 0.58*(KhCO2*0.00035*1000 - S(10));
MT(4) = 0;
MT(5) = 0;
MT(6) = 0;

% Substrate feed concentrations
Sfeed(1) = 15.01;
Sfeed(2) = KhO2*0.21*1000;
Sfeed(3) = KhCO2*0.00035*1000;
Sfeed(4) = 0;
Sfeed(5) = 40;
Sfeed(6) = 0; % The elements of the flux matrix have the sign given to
them by the
% coefficients in the Cost vector in main.
% Example, if:
% C{k}(i).rxns = [144, 832, 931];
% C{k}(i).wts = [3, 1, -1];
% Then the cost vector for this LP will be:

```

```

% flux(k,i) = 3*v_144 + v_832 - v_931 %% Update bounds and solve
for fluxes
[flux,penalty] = solveModel(t,y,INFO);

% Yeast fluxes
for i=1:4
v(1,i) = flux(1,i+1);
end
v(1,5) = 0;
v(1,6) = 0;

% Algae fluxes
v(2,1) = 0;
v(2,2) = flux(2,3);
v(2,3) = flux(2,4);
v(2,4) = 0;
v(2,5) = flux(2,2);

%% Dynamics dy(1) = Fin-Fout; % Volume
dy(2) = flux(1,1)*y(2) + (Xfeed(1)*Fin - y(2)*Fout)/y(1);
% Biomass yeast
dy(3) = flux(2,1)*y(3) + (Xfeed(2)*Fin - y(3)*Fout)/y(1);
% Biomass algae
for i = 1:5
dy(i+3) = v(1,i)*X(1) + v(2,i)*X(2) + MT(i) +
(Sfeed(i)*Fin - S(i)*Fout)/y(1);
end

if (S(2)/1000 > KhO2 && dy(3+2)>0)
dy(3+2) = 0;

```

```

end

if (S(3)/1000 > KhCO2 && dy(3+3)>0)
dy(3+3) = 0;
end

dy(9) = 0; % Leave total nitrogen constant
dy(10) = penalty(1) + penalty(2); %Penalty function

% Algebraic Equations
Ka = 10-9.4003;
K1c = 10-6.3819;
K2c = 10-10.3767;
Nt = S(6);
Ct = S(3);

x = y(11:16);
F = [-x(1) + Nt*x(6)/(x(6) + Ka) ;
-x(4) + x(1)/(1 + 2*K2c/x(6));
-x(3) + x(6)*x(4)/K1c;
-x(5) + x(4)*K2c/x(6);
-Nt + x(1) + x(2)
-Ct + x(3) + x(4) + x(5)];
dy(11:16) = F;

end

```

A.2.6 Inputs for the RHS.m file

The RHS function defined in RHS.m takes time, states, and the INFO structure and returns two matrices containing the upper and lower bounds for the fluxes specified

in the *exID* cell in main.m.

```
function [lb,ub] = RHS( t,y,INFO )
```

Here, *lb* corresponds to the lower bounds and *ub* to the upper bounds. Element *lb(i,j)* contains the lower bound corresponding to flux *j* in *exID{i}*. The same indexing applies for the *ub* matrix containing the upper bounds. The lower and upper bound quantities are functions of time and states. These functions must be continuous functions. In addition, if any lower bound or upper bound is defined as infinity or minus infinity, the value of this bound must remain constant the entire time of simulation.

A.2.7 Sample inputs for the RHS.m file

```
function [lb,ub] = RHS( t,y,INFO )
```

```
% Y1 = Volume (L)
% Y2 = Biomass Yeast (gDW/L)
% Y3 = Biomass Algae (gDW/L)
% Y4 = Glucose (mmol/L)
% Y5 = O2 (mmol/L)
% Y6 = Total Carbon (mmol/L)
% Y7 = Ethanol (mmol/L)
% Y8 = Acetate (mmol/L)
% Y9 = Total Nitrogen (mmol/L)
% Y10 = Penalty
% Y11 = NH4+
% Y12 = NH3
% Y13 = CO2
% Y14 = HCO3-
% Y15 = CO3 -2
% Y16 = H +
```

```

% This subroutine updates the upper and lower bounds for the
fluxes in the
% exID arrays in main. The output should be two matrices, lb and ub.
The lb matrix
% contains the lower bounds for exID{i} in the ith row in the same
order as
% exID. The same is true for the upper bounds in the ub matrix.
% Infinity can be used for unconstrained variables, however, it should
be
% fixed for all time.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Yeast bounds
% Glucose
if (y(4)<0)
lb(1,1) = 0;
else
lb(1,1) = -(20*y(4)/(0.5/0.18 + y(4)))*1/(1+y(7)/(10/0.046));
end
ub(1,1) = 0;
% Oxygen
lb(1,2) = -8*y(5)/(0.003/0.016 + y(5));
ub(1,2) = 0;
% CO2
lb(1,3) = 0;
ub(1,3) = Inf;
% Ethanol
lb(1,4) = 0;
ub(1,4) = Inf;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % Algae bounds
% HCO3

```

```

lb(2,1) = 0;
ub(2,1) = 0;
% Acetate
lb(2,2) = -14.9*y(8)/(2.2956+y(8)+y(8)^2/0.1557);
ub(2,2) = 0;
% Oxygen
lb(2,3) = -1.41750070911*y(5)/(0.009+y(5));
ub(2,3) = Inf;
% CO2
lb(2,4) = -2.64279793224*y(13)/(0.0009+y(13));
ub(2,4) = Inf;
% Light
Ke1 = 0.32;
Ke2 = 0.03;
L = 0.4; % meters depth of pond
Ke = Ke1 + Ke2*(y(3)+y(2)/2);
Io = 28*(max(sin(2*pi()*t/48)^2, sin(2*pi()*5/48)^2)
-sin(2*pi()*5/48)^2)/(1-sin(2*pi()*5/48)^2);
lb(2,5) = 0;
ub(2,5) = Io*(1-exp(-L*Ke))/(Ke*L);
% Other possible light sources set equal to zero
for i=6:16
lb(2,i) = 0;
ub(2,i) = 0;
end
% H+
lb(2,17) = -10;
ub(2,17) = Inf;
% Autotrophic growth
lb(2,18) = 0;

```

```

ub(2,18) = 0;
% Mixotrophic growth
lb(2,19) = 0;
ub(2,19) = Inf;
% Heterotrophic growth
lb(2,20) = 0;
ub(2,20) = 0;
% Non-growth associated ATP maintenance
lb(2,21) = 0.183;
ub(2,21) = 0.183;
% Starch
lb(2,22) = 0;
ub(2,22) = 0;
% Photoevolved oxygen
lb(2,23) = 0;
ub(2,23) = 8.28;
% Ethanol
lb(2,24) = 0;
ub(2,24) = 0;
end

```

A.2.8 Inputs for the `evts.m` file

This file is critical for DFBAlab to perform efficient DFBA simulations and will most likely not require any changes. This file needs to be changed only if event detection is needed in addition to the event detection associated with the LPs embedded. In this case, any event detection conditions can be added at the end of the vectors *value*, *isterminal* and *direction*. The definition of these vectors can be found in the MATLAB documentation.

```

function [value,isterminal,direction] = evts(t,y,INFO)
eps = INFO.tolevt;

```

```

lexID = INFO.lexID;
nmodel = INFO.nmodel;
bmodel = INFO.b;
lbct = INFO.lbct;
indlb = INFO.indlb;
indub = INFO.indub;
U = INFO.U;
L = INFO.L;
P = INFO.P;
Q = INFO.Q;
%% Update solutions
[lbx,ubx] = RHS( t,y,INFO );
ct = 0;
total = 0;
for i=1:nmodel
total = length(bmodel{i}) + total;
end
value = zeros(total,1);
isterminal = ones(total,1); % stop the integration
direction = -1*ones(total,1); % negative direction
for i=1:nmodel
b = bmodel{i};
lb = lbx(i,1:lexID(i));
ub = ubx(i,1:lexID(i));
lb(indlb{i}) = [];
ub(indub{i})=[];
b(1:length(lb)) = lb;
b(length(lb)+lbct(i)+1:length(lb)+lbct(i)+length(ub)) = ub;
x = (L{i} \ (P{i}*b));
x = U{i} \ x;

```

```

x = Q{i}*x;
% Detect when a basic variable crosses zero.
value(1+ct:length(x)+ct) = x + eps;
ct = ct + length(x);
end
ADD NEW CONDITIONS HERE
end

```

If the code is not modified, the length of the vectors *value*, *isterminal*, and *direction* is equal to the sum of all m_i where m_i corresponds to the number of rows of $model\{i\}.A$. In addition, the `main.m` file needs to be modified at the integration loop to distinguish events due to basis changes in the LPs from other types of events. In Example 3 in [44], this file is not modified.

A.3 Notes

Note 1: In some instances, DFBAlab may fail due to infeasible or unbounded LPs. Although theoretically this should not happen, it does happen some times numerically. When encountering this problem, change `INFO.tol` a bit and/or change the LP solver and try again.

Note 2: DFBAlab is designed such that only the fluxes defined as cost vectors can be accessed at `DRHS.m`. This is to ensure uniqueness in the right-hand side of the ODE/DAE system. Future versions of DFBAlab may contain the option of extracting all other fluxes, although these other fluxes must be treated carefully as they can be nonunique.

Appendix B

Extensions of Proposition 4.12 in Bonnans and Shapiro [14]

Proposition 4.12 in [14] refers to optimization problems of the form in (6.9) and requires Definition 6.4.1.

Proposition B.0.1. (Proposition 4.12 in [14].) Suppose that

1. the function $f(\mathbf{x}, \mathbf{u})$ is continuous on $X \times U$,
2. the inf-compactness condition holds at \mathbf{u}_0 ,
3. for any $\mathbf{x} \in \Phi$ the function $f_{\mathbf{x}}(\cdot) := f(\mathbf{x}, \cdot)$ is directionally differentiable at \mathbf{u}_0 ,
4. if $\mathbf{d} \in U$, $t_n \downarrow 0$ and $\{\mathbf{x}_n\}$ is a sequence in C , then $\{\mathbf{x}_n\}$ has a limit point $\bar{\mathbf{x}}$ such that

$$\limsup_{n \rightarrow \infty} \frac{f(\mathbf{x}_n, \mathbf{u}_0 + t_n \mathbf{d}) - f(\mathbf{x}_n, \mathbf{u}_0)}{t_n} \geq f'_{\bar{\mathbf{x}}}(\mathbf{u}_0; \mathbf{d}). \quad (\text{B.1})$$

Then the optimal value function $v(\mathbf{u})$ is directionally differentiable at \mathbf{u}_0 and

$$v'(\mathbf{u}_0; \mathbf{d}) = \inf_{\mathbf{x} \in S(\mathbf{u}_0)} f'_{\mathbf{x}}(\mathbf{u}_0; \mathbf{d}), \quad (\text{B.2})$$

where $S(\mathbf{u}) := \arg \min_{\mathbf{x} \in \Phi} f(\mathbf{x}, \mathbf{u})$. Moreover, if $\mathbf{x}_n \in S(\mathbf{u}_0 + t_n \mathbf{d})$ for some $t_n \downarrow 0$, then any limit point $\bar{\mathbf{x}}$ of $\{\mathbf{x}_n\}$ belongs to $S_1(\mathbf{u}_0, \mathbf{d})$, where $S_1(\mathbf{u}_0, \mathbf{d}) := \arg \min_{\mathbf{x} \in S(\mathbf{u}_0)} f'_\mathbf{x}(\mathbf{u}_0, \mathbf{d})$.

Next, we extend Proposition 4.12 in [14] to calculate LD-derivatives of optimization problems of the form in (6.9). To do this, we apply Proposition 4.12 recursively. Not all PC^1 functions are suitable for LD-derivative computation using Proposition 4.12 in [14] as the following example shows.

Example B.0.1. Consider the objective function $f(x, u) = |x - u|$. f is clearly a PC^1 function on $\mathbb{R} \times \mathbb{R}$. Then if $d > 0$,

$$f'(x, u; 0, d) = \begin{cases} -d & \text{if } x \geq u \\ d & \text{if } x < u \end{cases}$$

and if $d < 0$, $f'(x, u; 0, d) = -f'(x, u; 0, -d)$. For any $d \neq 0$, $f'(x, u; 0, d)$ will be discontinuous at $x = u$. Therefore, $f'(\cdot, u; 0, \cdot)$ is not a continuous function on $\mathbb{R} \times \mathbb{R}$ and Proposition 4.12 in [14] cannot be applied recursively.

Now consider $f(x, u) = x + |u|$. Then for $d > 0$,

$$f'(x, u; 0, d) = \begin{cases} -d & \text{if } u < 0 \\ d & \text{if } u \geq 0 \end{cases}$$

and if $d < 0$, $f'(x, u; 0, d) = -f'(x, u; 0, -d)$. Therefore, $f'(\cdot, u; 0, \cdot)$ is continuous on $\mathbb{R} \times \mathbb{R}$ and Proposition 4.12 in [14] can be applied recursively to compute LD-derivatives. Therefore for an objective function $f : X \times U \rightarrow \mathbb{R}$, nonsmoothness can be tolerated on U but not on X when computing LD-derivatives.

The following example illustrates how for a PC^1 function $f : X \subset \mathbb{R}^x \rightarrow \mathbb{R}$ with selection functions f^1, \dots, f^{n_f} , a point $\mathbf{x} \in X$ and a direction $\mathbf{d} \in \mathbb{R}^x$, there may not exist $\delta > 0$ such that for all $\epsilon \in [0, \delta)$, there exists $i \in \{1, \dots, n_f\}$ such that $f(\mathbf{x} + \epsilon \mathbf{d}) = f^i(\mathbf{x} + \epsilon \mathbf{d})$.

Example B.0.2. Consider the two following selection functions $f^1, f^2 : \mathbb{R} \rightarrow \mathbb{R}$:

$$f^1(u) = \begin{cases} 0 & \text{if } u = 0, \\ u^3 \sin(1/u) & \text{else.} \end{cases}, \quad f^2(u) = \begin{cases} 0 & \text{if } u = 0, \\ u^3 \cos(1/u) & \text{else.} \end{cases},$$

and let $f(u) = \max\{f^1(u), f^2(u)\}$. Both f^1, f^2 are C^1 functions. To see this, consider that for any sequence $\{u_n\}$ where $u_n \rightarrow 0$

$$\begin{aligned} \lim_{n \rightarrow \infty} f^1(u_n) &= \lim_{n \rightarrow \infty} u_n^3 \sin(1/u_n), \\ |f^1(u_n)| &= |u_n^3 \sin(1/u_n)| \leq |u_n^3|, \\ \lim_{n \rightarrow \infty} |u_n^3| &= 0, \\ \lim_{n \rightarrow \infty} f^1(u_n) &= \lim_{n \rightarrow \infty} u_n^3 \sin(1/u_n) = 0, \end{aligned}$$

and by a similar argument $\lim_{n \rightarrow \infty} f^2(u_n) = 0$. Therefore, both f^1, f^2 are continuous at $\{0\}$. For this functions to be C^1 , we must verify that their derivative is continuous at $\{0\}$ too. First, we compute the derivative of f^1 and f^2 from the definition. Then,

$$[f^1]'(0) = \lim_{t \rightarrow 0} \frac{f^1(t) - 0}{t - 0} = \lim_{t \rightarrow 0} \frac{t^3 \sin(1/t)}{t} = \lim_{t \rightarrow 0} t^2 \sin(1/t) = 0,$$

and by a very similar argument $[f^2]'(0) = 0$. For $u \neq 0$, $[f^1]'(u) = 3u^2 \sin(1/u) - u \cos(1/u)$ and $[f^2]'(u) = 3u^2 \cos(1/u) + u \sin(1/u)$ and

$$\lim_{u \rightarrow 0} [f^1]'(u) = \lim_{u \rightarrow 0} [f^2]'(u) = 0,$$

which establishes continuity of both $[f^1]', [f^2]'$ at $\{0\}$. However, notice that for any $d \neq 0$, it is impossible to find a $\delta > 0$ such that for all $\epsilon \in [0, \delta)$, $f(\epsilon d) = f^i(\epsilon d)$ with $i \in \{1, 2\}$ because near zero, an infinite number of switches between f^1 and f^2 take place.

Theorem B.0.2. Let $\mathbf{M} \in \mathbb{R}^{u \times q}$. Suppose that

1. the function $f(\mathbf{x}, \mathbf{u})$ is PC^1 on $X \times U$,

2. the function $f(\cdot, \mathbf{u})$ is C^2 on X for every $\mathbf{u} \in U$,
3. the inf-compactness condition holds at \mathbf{u}_0 .

Then the optimal value function v is l -smooth at \mathbf{u}_0 and for $i = \{0, \dots, q\}$

$$v_{\mathbf{u}_0, \mathbf{M}}^{(i)}(\mathbf{d}) = \min_{\mathbf{x} \in S_i(\mathbf{u}_0, \mathbf{M})} [f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(i)}(\mathbf{d}) \quad (\text{B.3})$$

where $S_0(\mathbf{u}_0, \mathbf{M}) = S(\mathbf{u}_0)$ and for $i = \{1, \dots, q\}$

$$S_i(\mathbf{u}_0, \mathbf{M}) = \arg \min_{\mathbf{x} \in S_{i-1}(\mathbf{u}_0, \mathbf{M})} [f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(i-1)}(\mathbf{m}_i). \quad (\text{B.4})$$

Proof: The strategy of this proof is to apply Proposition B.0.1 recursively. Since the function $f_{\mathbf{x}}(\cdot) \equiv f(\mathbf{x}, \cdot)$ is PC^1 , it is l -smooth and directionally differentiable. Therefore, the first three assumptions in Proposition B.0.1 are satisfied.

Next, we verify the last assumption in Proposition B.0.1. If f is continuously differentiable at $(\bar{\mathbf{x}}, \mathbf{u}_0)$, then, it is strictly differentiable and the generalized directional derivative coincides with the directional derivative (Definition 2.3.4 and Proposition 2.3.6 in [27]). Consider the sequence $\{\mathbf{u}_n\}$ with $\mathbf{u}_n \rightarrow \mathbf{u}_0$. Then,

$$\begin{aligned} f^\circ(\bar{\mathbf{x}}, \mathbf{u}_0; \mathbf{0}, \mathbf{d}) &= \limsup_{n \rightarrow \infty} \frac{f(\mathbf{x}_n, \mathbf{u}_n + t_n \mathbf{d}) - f(\mathbf{x}_n, \mathbf{u}_n)}{t_n} \\ &\geq \limsup_{n \rightarrow \infty} \frac{f(\mathbf{x}_n, \mathbf{u}_0 + t_n \mathbf{d}) - f(\mathbf{x}_n, \mathbf{u}_0)}{t_n} \\ &\geq \lim_{n \rightarrow \infty} \frac{f(\bar{\mathbf{x}}, \mathbf{u}_0 + t_n \mathbf{d}) - f(\bar{\mathbf{x}}, \mathbf{u}_0)}{t_n} \\ &= f'(\bar{\mathbf{x}}, \mathbf{u}_0; \mathbf{0}, \mathbf{d}) = f'_{\bar{\mathbf{x}}}(\mathbf{u}_0; \mathbf{d}) = f^\circ(\bar{\mathbf{x}}, \mathbf{u}_0; \mathbf{0}, \mathbf{d}). \end{aligned} \quad (\text{B.5})$$

Since f is PC^1 , it has a finite number of C^1 selection functions n_f that can be labeled f^1, \dots, f^{n_f} . Since $f(\cdot, \mathbf{u})$ is a C^2 function, the pieces visited by the sequence in the limit in (B.1) are just a function of the sequence $\{t_n\}$ and not of the sequence $\{\mathbf{x}_n\}$. If for any sequence $\{t_n\}$, there exists $N \in \mathbb{N}$ such that for all $n > N$ there exists $i \in \{1, \dots, n_f\}$ such that $f^i(\bar{\mathbf{x}}, \mathbf{u}_0 + t_n \mathbf{d}) = f(\bar{\mathbf{x}}, \mathbf{u}_0 + t_n \mathbf{d})$, then all elements

in the sequence can be described by a C^1 function and the relationships in (B.5) hold. If there exists more than one selection function f^i with $i \in J(\mathbf{d}) \subset \{1, \dots, n_f\}$ that must be considered (see Example B.0.2), then there exist subsequences $\{t_n^i\}$ such that all elements in the subsequence can be described by the function f^i with $i \in J(\mathbf{d})$. Then again, the relationships in (B.5) hold and we obtain the limits $[f^i]_{\bar{\mathbf{x}}}(\mathbf{u}_0; \mathbf{d})$. However, since $f(\mathbf{x}, \cdot)$ is l -smooth for all $\mathbf{x} \in X$, then for all $i \in J(\mathbf{d})$, $[f^i]_{\bar{\mathbf{x}}}(\mathbf{u}_0; \mathbf{d}) = f'_{\bar{\mathbf{x}}}(\mathbf{u}_0; \mathbf{d})$. This implies that the last assumption in Proposition B.0.1 is satisfied and therefore,

$$v_{\mathbf{u}_0, \mathbf{M}}^{(0)}(\mathbf{d}) = \inf_{\mathbf{x} \in S_0(\mathbf{u}_0, \mathbf{M})} [f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(0)}(\mathbf{d}). \quad (\text{B.6})$$

From the inf-compactness condition we get $S_0(\mathbf{u}_0, \mathbf{M})$ must be bounded and from the continuity of f we get that $S_0(\mathbf{u}_0, \mathbf{M})$ must be closed; it follows that $S_0(\mathbf{u}_0, \mathbf{M})$ can be described as a compact set or as the union of disjoint compact sets. Let $f_{\mathbf{u}_0, \mathbf{M}}^{[0]}(\mathbf{x}, \mathbf{d}) \equiv [f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(0)}(\mathbf{d})$. $f_{\mathbf{u}_0, \mathbf{M}}^{[0]}$ is a PC^1 function. To see this, notice that

$$f'_{\bar{\mathbf{x}}}(\mathbf{u}_0; \mathbf{d}) = \nabla_{\mathbf{u}} f^i(\bar{\mathbf{x}}, \mathbf{u}_0)[\mathbf{d}],$$

for some $i \in \{1, \dots, n_f\}$. Since $f(\cdot, \mathbf{u})$ is a C^2 function for all $\mathbf{u} \in U$, i is only dependent on \mathbf{d} . Therefore for all $\mathbf{x} \in X$, $f_{\mathbf{u}_0, \mathbf{M}}^{[0]}(\mathbf{x}, \cdot)$ is piecewise linear. In addition for all i and all $\mathbf{u} \in U$, $\nabla_{\mathbf{u}} f^i(\cdot, \mathbf{u})$ are C^1 functions. Since $f_{\mathbf{u}_0, \mathbf{M}}^{[0]}$ can be described as the composition of a piecewise linear function with C^1 functions, then $f_{\mathbf{u}_0, \mathbf{M}}^{[0]}$ is continuous. Therefore, the minimum in (B.6) is attained. To compute the rest of the LD-derivatives, we must verify whether the optimization problem

$$v_{\mathbf{u}_0, \mathbf{M}}^{(0)}(\mathbf{d}_0) = \min_{\mathbf{x} \in S_0(\mathbf{u}_0, \mathbf{M})} f_{\mathbf{u}_0, \mathbf{M}}^{[0]}(\mathbf{x}, \mathbf{d}_0) \quad (\text{B.7})$$

satisfies the Assumptions in Proposition B.0.1. Since $S_0(\mathbf{u}_0, \mathbf{M})$ can be described as a compact set or the union of disjoint compact sets, the inf-compactness condition holds. Assumptions (i) and (iii) follow from the continuity of $f_{\mathbf{u}_0, \mathbf{M}}^{[0]}$ and the l -smoothness of $f_{\mathbf{x}}$ for all $\mathbf{x} \in X$. Since $f_{\mathbf{u}_0, \mathbf{M}}^{[0]}$ is continuous and the inf-compactness condition holds,

$S_1(\mathbf{u}_0, \mathbf{M})$ can be described as a compact set or the union of disjoint compact sets. To verify Assumption (iv), notice again that which selection functions are active depend exclusively on the $\{t_n\}$ sequence and not on the $\{\mathbf{x}_n\}$ sequence. Consider that the index set $K(\mathbf{d}) \subset J(\mathbf{m}_1)$ contains the selection functions for which subsequences $\{t_n^i\}$ can be constructed such that all the subsequence elements stay in selection function $i \in K(\mathbf{d})$. To see that the limit in (B.1) converges, consider these subsequences $\{t_n^i\}$ and let $\mathbf{d}_0 = \mathbf{m}_1$, $\mathbf{d} \in \mathbb{R}^u$,

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{f_{\mathbf{u}_0, \mathbf{M}}^{[0]}(\mathbf{x}_n, \mathbf{m}_1 + t_n^i \mathbf{d}) - f_{\mathbf{u}_0, \mathbf{M}}^{[0]}(\mathbf{x}_n, \mathbf{m}_1)}{t_n^i} = \\ \lim_{n \rightarrow \infty} \frac{\nabla_{\mathbf{u}} f^i(\mathbf{x}_n, \mathbf{u}_0)[\mathbf{m}_1 + t_n^i \mathbf{d}] - \nabla_{\mathbf{u}} f^i(\mathbf{x}_n, \mathbf{u}_0)[\mathbf{m}_1]}{t_n^i} = \nabla_{\mathbf{u}} f^i(\bar{\mathbf{x}}, \mathbf{u}_0)[\mathbf{d}] = [f_{\mathbf{u}_0, \mathbf{M}}^{[0]}]_{\bar{\mathbf{x}}}(\mathbf{m}_1; \mathbf{d}), \end{aligned}$$

for each $i \in K(\mathbf{d})$. Since $f(\mathbf{x}, \cdot)$ is l -smooth for all $\mathbf{x} \in X$, $\nabla_{\mathbf{u}} f^i(\bar{\mathbf{x}}, \mathbf{u}_0)[\mathbf{d}] = [f_{\mathbf{u}_0, \mathbf{M}}^{[0]}]_{\bar{\mathbf{x}}}(\mathbf{m}_1; \mathbf{d})$ for all $i \in K(\mathbf{d})$, and the limit in Assumption (iv) in Proposition B.0.1 exists. Therefore,

$$v_{\mathbf{u}_0, \mathbf{M}}^{(1)}(\mathbf{d}) = \inf_{\mathbf{x} \in S_1(\mathbf{u}_0, \mathbf{M})} [f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(1)}(\mathbf{d}) = \inf_{\mathbf{x} \in S_1(\mathbf{u}_0, \mathbf{M})} f_{\mathbf{u}_0, \mathbf{M}}^{[1]}(\mathbf{x}, \mathbf{d}). \quad (\text{B.8})$$

To finish the proof, we need to show that $[f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(1)}$ is a PC^1 function, that $[f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(1)}(\cdot, \mathbf{d})$ is a C^1 function for any $\mathbf{d} \in \mathbb{R}$, and that $S_2(\mathbf{u}_0, \mathbf{M})$ can be described as a compact set or the union of compact sets. It is clear that for each $\mathbf{x} \in X$, $[f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(1)}(\mathbf{x}, \cdot)$ is a piecewise linear function with selection functions $\nabla_{\mathbf{u}} f^i(\mathbf{x}, \mathbf{u}_0)[\cdot]$ for $i \in \{1, \dots, n_f\}$. For all i and all $\mathbf{u}_0 \in U$, $\nabla_{\mathbf{u}} f^i(\cdot, \mathbf{u}_0)$ is a C^1 function. Therefore, $[f_{\mathbf{x}}]_{\mathbf{u}_0, \mathbf{M}}^{(1)}$ is continuous because it results from the composition of a piecewise linear function with C^1 functions. Then, the minimum in (B.8) is attained, $S_2(\mathbf{u}_0, \mathbf{M})$ can be described as a compact set or the union of compact sets, and

$$v_{\mathbf{u}_0, \mathbf{M}}^{(1)}(\mathbf{d}) = \min_{\mathbf{x} \in S_1(\mathbf{u}_0, \mathbf{M})} f_{\mathbf{u}_0, \mathbf{M}}^{[1]}(\mathbf{x}, \mathbf{d}_0). \quad (\text{B.9})$$

Since (B.9) has the same structure as (B.7), Proposition B.0.1 applies to (B.9), and the rest of the LD-derivatives can be computed. The proof follows by induction. \square

Appendix C

Convex and Concave Relaxations of Linear Programs and Lexicographic Linear Programs.

As presented in Chapter 8, global optimization algorithms are necessary to find a global optimum or a solution with an objective function value within ϵ of the global minimum. In this context, convex and concave relaxations are required. In particular, these relaxations are necessary to compute reliably lower bounds for the subproblems generated in the branch and bound algorithm. The aim of this Appendix is to construct convex and concave relaxations of factorable functions with a linear program (LP) embedded.

C.1 Preliminaries

Standard McCormick relaxations can be generated for factorable functions (see [92]). However, this procedure cannot be applied to generate convex and concave relaxations of lexicographic LPs (LLPs) because obtaining the factored representation of an LP is computationally intractable. Therefore, generalized McCormick relaxations, presented in [121], are needed.

Generalized McCormick relaxations enable the use of McCormick relaxations in

more complex settings, by flexibilizing the initialization rules of the process generating the convex and concave relaxations. The first step in generating standard McCormick relaxations requires setting $v_i^{cv}(\mathbf{p}) = v_i^{cc}(\mathbf{p}) = p_i$ for $i = 1, \dots, n_p, \forall \mathbf{p} \in P$. Generalized McCormick relaxations take $v_i^{cv/cc}$ as arguments separate from the value of \mathbf{p} . Then, previously constructed relaxations can be used as inputs when composing with later factors. Proposition 3 in [121] asserts the validity of generalized convex and concave McCormick relaxations when a more general initialization of the $v_i^{cv/cc}$ factors is used. In particular, this proposition ensures that if convex and concave relaxations are generated for LLPs, then Definition 15 in [121] allows the computation of McCormick relaxations for compositions involving LLPs.

C.1.1 McCormick's composition theorem

McCormick's univariate composition theorem is presented in Definition 9 in [121], which is taken from [92]. Consider a convex set $C \in \mathbb{R}^n$. This theorem allows to compute relaxations for $\varphi \circ f$ where $f : C \rightarrow \mathbb{R}$ is bounded on C , $f(C) \subset [a, b]$ and $\varphi : [a, b] \rightarrow \mathbb{R}$ when convex and concave relaxations of φ on $[a, b]$ are known. However, LLPs parameterized by their right-hand side take a multivariate input. Therefore, a multivariate version of McCormick's composition theorem is needed to compute relaxations of LLPs.

First, we introduce the following Proposition.

Proposition C.1.1. Let C be a nonempty, bounded convex subset of \mathbb{R}^n . Let $g^{cv} : C \rightarrow \mathbb{R}$ be a convex function and $g^{cc} : C \rightarrow \mathbb{R}$ a concave function. Then g^{cv} is bounded below and g^{cc} is bounded above.

Proof: The following definitions and theorems need to be introduced.

Definition C.1.1. The convex function $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ is said to be proper if $f(\mathbf{x}) < +\infty$ for at least one $\mathbf{x} \in \mathbb{R}^n$ and $f(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in \mathbb{R}^n$.

Definition C.1.2. The relative interior of a convex set $C \subset \mathbb{R}^n$, denoted $\text{ri}(C)$, is the interior of C regarded as a subset of its affine hull.

Definition C.1.3. A vector $\mathbf{s} \in \mathbb{R}^n$ is a subgradient of a convex function f at \mathbf{x} if

$$f(\mathbf{x}') \geq f(\mathbf{x}) + \mathbf{s}^T(\mathbf{x}' - \mathbf{x}), \quad \forall \mathbf{x}' \in \mathbb{R}^n.$$

The set of all subgradients of f at \mathbf{x} is called the subdifferential at \mathbf{x} and denoted $\partial f(\mathbf{x})$.

Theorem C.1.2. (Theorem 6.2 in [110]). Let C be a nonempty convex subset of \mathbb{R}^n . Then, its relative interior $\text{ri}(C)$ is nonempty.

Theorem C.1.3. (Theorem 23.4 in [110]). Let f be a proper convex function. For $\mathbf{x} \in \text{ri}(\text{dom } f)$, $\partial f(\mathbf{x})$ is nonempty.

Proof of Proposition C.1.1: To be precise, define $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ by $f(\mathbf{x}) = g^{cv}(\mathbf{x})$ for $\mathbf{x} \in C$, $f(\mathbf{x}) = +\infty$ for $\mathbf{x} \notin C$. Then f is a proper convex function and $\text{dom } f = C$. By Theorem C.1.2, the relative interior of C is nonempty, and so by Theorem C.1.3, there exists $\mathbf{x} \in \text{ri}(C)$ at which $\partial f(\mathbf{x})$ is nonempty. Consequently, there exists $\mathbf{s} \in \mathbb{R}^n$ such that

$$f(\mathbf{x}') \geq f(\mathbf{x}) + \mathbf{s}^T(\mathbf{x}' - \mathbf{x}), \quad \forall \mathbf{x}' \in \mathbb{R}^n.$$

Since C is bounded, there exists $k \in \mathbb{R}$ such that $\|\mathbf{x}' - \mathbf{x}\|_2 \leq k$ for all $\mathbf{x}' \in C$. Thus $|\mathbf{s}^T(\mathbf{x}' - \mathbf{x})| \leq k\|\mathbf{s}\|_2$. Since $f(\mathbf{x}) + \mathbf{s}^T(\mathbf{x}' - \mathbf{x}) \geq f(\mathbf{x}) - |\mathbf{s}^T(\mathbf{x}' - \mathbf{x})|$, we have

$$f(\mathbf{x}') \geq f(\mathbf{x}) - k\|\mathbf{s}\|_2$$

for all $\mathbf{x}' \in C$. Finally, we note that this means $g^{cv}(\mathbf{x}') \geq g^{cv*}$, for all $\mathbf{x}' \in C$. The same reasoning applies for g^{cc} . Notice that $-g^{cc}$ is a convex function. Then, $-g^{cc}$ is bounded below which implies g^{cc} is bounded above. \square

The following Theorem is based on Theorem 2 in [130] under weaker assumptions.

Theorem C.1.4. Let $C \subset \mathbb{R}^n$ and $Z \subset \mathbb{R}^m$ be nonempty convex sets. Consider the composite function $g \equiv \varphi \circ \mathbf{f}$ where $\mathbf{f} : C \rightarrow \mathbb{R}^m$ with $\mathbf{f}(C) \subset Z$ and $\varphi : Z \rightarrow \mathbb{R}$.

Suppose that convex functions $f_i^{cv} : C \rightarrow \mathbb{R}$ and concave functions $f_i^{cc} : C \rightarrow \mathbb{R}$ satisfying

$$f_i^{cv}(\mathbf{x}) \leq f_i(\mathbf{x}) \leq f_i^{cc}(\mathbf{x}), \forall \mathbf{x} \in C, \forall i = 1, \dots, m$$

are known. Let $\varphi^{cv} : Z \rightarrow \mathbb{R}$ and $\varphi^{cc} : Z \rightarrow \mathbb{R}$ be convex and concave relaxations of φ on Z . Then the following functions are well-defined and are convex and concave relaxations of g on C :

$$u : C \rightarrow \mathbb{R} : \mathbf{x} \mapsto \inf_{\mathbf{z} \in Z} \{\varphi^{cv}(\mathbf{z}) : \mathbf{f}^{cv}(\mathbf{x}) \leq \mathbf{z} \leq \mathbf{f}^{cc}(\mathbf{x})\}, \quad (\text{C.1})$$

$$o : C \rightarrow \mathbb{R} : \mathbf{x} \mapsto \sup_{\mathbf{z} \in Z} \{\varphi^{cc}(\mathbf{z}) : \mathbf{f}^{cv}(\mathbf{x}) \leq \mathbf{z} \leq \mathbf{f}^{cc}(\mathbf{x})\}. \quad (\text{C.2})$$

Proof: For any $\mathbf{x} \in C$, the inequalities $\mathbf{f}^{cv}(\mathbf{x}) \leq \mathbf{z} \leq \mathbf{f}^{cc}(\mathbf{x})$ define a compact set $D(\mathbf{x})$. $Z \cap D(\mathbf{x})$ is nonempty because $\mathbf{f}(\mathbf{x}) \in D(\mathbf{x}), \mathbf{f}(\mathbf{x}) \in Z$, thus $o(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in C$. Z is a convex set and $D(\mathbf{x})$ is a compact set, then $Z \cap D(\mathbf{x})$ is convex and bounded for all $\mathbf{x} \in C$. Then from Proposition C.1.1, φ^{cc} is bounded above on $Z \cap D(\mathbf{x})$, which means that $o(\mathbf{x})$ must be finite. Then $o(\mathbf{x})$ is well-defined.

For any $\mathbf{x} \in C, \mathbf{f}(\mathbf{x}) \in Z$. By definition, $\mathbf{f}^{cv}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}) \leq \mathbf{f}^{cc}(\mathbf{x})$. Then

$$\varphi(\mathbf{f}(\mathbf{x})) \leq \varphi^{cc}(\mathbf{f}(\mathbf{x})) \leq \sup_{\mathbf{z} \in Z} \{\varphi^{cc}(\mathbf{z}) : \mathbf{f}^{cv}(\mathbf{x}) \leq \mathbf{z} \leq \mathbf{f}^{cc}(\mathbf{x})\} \equiv o(\mathbf{x}).$$

Then $o(\mathbf{x})$ is an upper bound on $\varphi(\mathbf{f}(\mathbf{x}))$, $\forall \mathbf{x} \in C$. Take $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y} = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$

and $\lambda \in (0, 1)$. Then,

$$\begin{aligned}
\mathbf{f}^{cv}(\mathbf{y}) &\leq \lambda \mathbf{f}^{cv}(\mathbf{x}_1) + (1 - \lambda) \mathbf{f}^{cv}(\mathbf{x}_2), \\
\mathbf{f}^{cc}(\mathbf{y}) &\geq \lambda \mathbf{f}^{cc}(\mathbf{x}_1) + (1 - \lambda) \mathbf{f}^{cc}(\mathbf{x}_2), \\
o(\mathbf{y}) &\equiv \sup_{\mathbf{z} \in Z} \{\varphi^{cc}(\mathbf{z}) : \mathbf{f}^{cv}(\mathbf{y}) \leq \mathbf{z} \leq \mathbf{f}^{cc}(\mathbf{y})\} \\
&\geq \sup_{\mathbf{z} \in Z} \{\varphi^{cc}(\mathbf{z}) : \mathbf{f}^{cv}(\mathbf{x}_1) + (1 - \lambda) \mathbf{f}^{cv}(\mathbf{x}_2) \leq \mathbf{z} \leq \lambda \mathbf{f}^{cc}(\mathbf{x}_1) + (1 - \lambda) \mathbf{f}^{cc}(\mathbf{x}_2)\}.
\end{aligned} \tag{C.3}$$

Consider the bounded convex sets $F_1 = [\mathbf{f}^{cv}(\mathbf{x}_1), \mathbf{f}^{cc}(\mathbf{x}_1)] \cap Z$ and $F_2 = [\mathbf{f}^{cv}(\mathbf{x}_2), \mathbf{f}^{cc}(\mathbf{x}_2)] \cap Z$, and the points $o(\mathbf{x}_1)$, $o(\mathbf{x}_2)$. Let $\{\widehat{\mathbf{z}}^{1,k}\}$ and $\{\widehat{\mathbf{z}}^{2,k}\}$ be sequences satisfying

$$\begin{aligned}
\{\widehat{\mathbf{z}}^{1,k}\} &\in Z \cap F_1, \forall k \in \mathbb{N}, \\
\{\widehat{\mathbf{z}}^{2,k}\} &\in Z \cap F_2, \forall k \in \mathbb{N}, \\
\lim_{k \rightarrow \infty} \varphi^{cv}(\widehat{\mathbf{z}}^{1,k}) &= o(\mathbf{x}_1), \text{ and} \\
\lim_{k \rightarrow \infty} \varphi^{cv}(\widehat{\mathbf{z}}^{2,k}) &= o(\mathbf{x}_2).
\end{aligned}$$

Such sequences are guaranteed to exist because, for example, $o(\mathbf{x}_1)$ is a limit point of the set $\{\varphi^{cc}(\mathbf{z}) : \mathbf{z} \in Z, \mathbf{f}^{cv}(\mathbf{x}_1) \leq \mathbf{z} \leq \mathbf{f}^{cc}(\mathbf{x}_2)\}$. Let $\widehat{\mathbf{z}}^{i,k}$ be an element of $Z \cap F_i$ such that $\varphi^{cc}(\widehat{\mathbf{z}}^{1,k}) \geq o(\mathbf{x}_1) + \frac{1}{k}$. For each k , let $\widehat{\mathbf{z}}^{y,k} \equiv \lambda \widehat{\mathbf{z}}^{1,k} + (1 - \lambda) \widehat{\mathbf{z}}^{2,k}$, which is feasible in (C.3) because

$$\begin{aligned}
\lambda \mathbf{f}^{cv}(\mathbf{x}_1) &\leq \lambda \widehat{\mathbf{z}}^{1,k} \leq \lambda \mathbf{f}^{cc}(\mathbf{x}_1), \\
(1 - \lambda) \mathbf{f}^{cv}(\mathbf{x}_2) &\leq (1 - \lambda) \widehat{\mathbf{z}}^{2,k} \leq (1 - \lambda) \mathbf{f}^{cc}(\mathbf{x}_2).
\end{aligned}$$

In addition $\widehat{\mathbf{z}}^{y,k} \in Z$ because it is a convex combination of two elements in Z which is a convex set. Then, from (C.3)

$$o(\mathbf{y}) \geq \varphi^{cc}(\widehat{\mathbf{z}}^{y,k}).$$

From concavity of φ^{cc} , we have

$$o(\mathbf{y}) \geq \varphi^{cc}(\widehat{\mathbf{z}}^{y,k}) \geq \lambda \varphi^{cc}(\widehat{\mathbf{z}}^{1,k}) + (1 - \lambda) \varphi^{cc}(\widehat{\mathbf{z}}^{2,k}).$$

Taking the limits,

$$\begin{aligned} o(\mathbf{y}) &\geq \lim_{k \rightarrow \infty} \varphi^{cc}(\widehat{\mathbf{z}}^{y,k}) \geq \lambda \lim_{k \rightarrow \infty} \varphi^{cc}(\widehat{\mathbf{z}}^{1,k}) + (1 - \lambda) \lim_{k \rightarrow \infty} \varphi^{cc}(\widehat{\mathbf{z}}^{2,k}) \\ &= \lambda o(\mathbf{x}_1) + (1 - \lambda) o(\mathbf{x}_2), \end{aligned}$$

so o is a concave overestimator of $\varphi \circ \mathbf{f}$. □

C.2 Convex and concave relaxations for LPs

Consider F as defined in Chapter 6 after Assumption 6.1.1 and \mathbf{h} as defined in (6.1) and (6.2). For simplicity, we shall define $h \equiv h_0$. This section discusses convex and concave relaxations for h on F . Let $B \subset F$ be a box (also known as m dimensional interval or m -cell). Theorem 5.1 in [12] establishes the convexity of h . Therefore, only a concave relaxation is needed.

In a one dimensional space, the concave envelope of a convex function on an interval is the secant going through the function value at the upper and lower bounds of the interval. However, finding a concave relaxation of a convex function in a higher dimensional space is not a trivial task. Theorem 2.4.2 in [37] provides a characterization of the convex envelope f_{env}^{cv} of a concave function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ over a

polytope $X := \text{conv}\{\mathbf{x}^1, \dots, \mathbf{x}^n\} \subset \mathbb{R}^n$:

$$\begin{aligned}
 f_{env}^{cv}(\mathbf{x}) &= \min_{\boldsymbol{\lambda} \in \mathbb{R}^n} \sum_{i=1}^n \lambda_i f(\mathbf{x}^i) \\
 \text{s.t.} \quad &\sum_{i=1}^n \lambda_i \mathbf{x}^i = \mathbf{x} \\
 &\sum_{i=1}^n \lambda_i = 1, \\
 &\lambda_i \geq 0, \quad \forall i = 1, \dots, n,
 \end{aligned} \tag{C.4}$$

for all $\mathbf{x} \in X$. This procedure takes the minimum of all convex combinations of the function value at the vertices of the polytope to construct the convex envelope. Therefore, a concave envelope of a convex function over a polytope can be constructed by taking the maximum of all convex combinations of the function value of the vertices of the polytope:

$$\begin{aligned}
 f_{env}^{cc}(\mathbf{x}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^n} \sum_{i=1}^n \lambda_i f(\mathbf{x}^i) \\
 \text{s.t.} \quad &\sum_{i=1}^n \lambda_i \mathbf{x}^i = \mathbf{x} \\
 &\sum_{i=1}^n \lambda_i = 1, \\
 &\lambda_i \geq 0, \quad \forall i = 1, \dots, n.
 \end{aligned} \tag{C.5}$$

for all $\mathbf{x} \in X$. An example implementing this expression to calculate the concave envelope of a simple LP can be found in Section C.6.1. To be able to construct a concave relaxation of an LP with respect to k variable right-hand sides, 2^k function evaluations are needed (all combinations of upper and lower bounds of variable right-hand sides b_i^L, b_i^U need to be considered). Each function evaluation involves the solution of a linear program. If this procedure is embedded within a branch and bound algorithm, everytime branching occurs, 2^{k-1} LPs will need to be solved before being able to use (C.5) to compute the concave envelope. In addition, the function evaluation of the

concave and convex envelopes involves solving (C.5) and (6.1), respectively. This can be a computationally expensive procedure.

Consider the dual problem of (6.1):

$$\begin{aligned} \max_{\boldsymbol{\mu} \in \mathbb{R}^m} \quad & \mathbf{b}^T \boldsymbol{\mu}, \\ \text{s.t.} \quad & \mathbf{A}^T \boldsymbol{\mu} \leq \mathbf{c}. \end{aligned} \tag{C.6}$$

To find an upper bound on h in (6.1) (UBD), the objective function of (C.6) needs to be maximized with respect to \mathbf{b} and $\boldsymbol{\mu}$:

$$\begin{aligned} \text{UBD} = \max_{\boldsymbol{\mu}, \mathbf{b} \in \mathbb{R}^m} \quad & \mathbf{b}^T \boldsymbol{\mu}, \\ \text{s.t.} \quad & \mathbf{A}^T \boldsymbol{\mu} \leq \mathbf{c}, \\ & b_i^L \leq b_i \leq b_i^U, \quad \forall i = 1, \dots, k. \end{aligned} \tag{C.7}$$

The inequalities $b_i^L \leq b_i \leq b_i^U$ define the box B . This NLP can be solved using BARON [114]. However, finding the value of h at the extreme points of B may be cheaper to compute than solving the NLP in BARON and provides enough information to compute an upper bound (for $\mathbf{b} \in B \subset \mathbb{R}^m$, only k components are variable and $m - k$ are fixed, therefore, only 2^k function evaluations are needed instead of 2^m).

Proposition C.2.1. Let $f : X \rightarrow \mathbb{R}$ be a continuous and convex function and X a convex and bounded polytope. Then, one of the extreme points of X attains the maximum of f over X .

Proof: Since $X \subset \mathbb{R}^n$ is a compact set, Weierstrass Theorem ensures that a point in X attains the maximum (Theorem 4.16 in [112]). All points in a polytope can be expressed as a convex combination of its extreme points. Assume that X has n_e extreme points. Then any $\mathbf{x} \in X$ can be written as:

$$\mathbf{x} = \sum_{i=1}^{n_e} \lambda_i \mathbf{x}_i,$$

where \mathbf{x}_i are the extreme points of X . For any $\mathbf{x} \in X$, consider the following proce-

dure:

1. First, order all λ_i from minimum to maximum. Delete all $\lambda_i = 0$ from the list. If only one element is in the list, then, the process is finished and $\mathbf{x} = \mathbf{x}_i$ where λ_i is the only member of the list. Assume there are p elements in the list.
2. Take the first two λ_a, λ_b . Make $\alpha_1 = \frac{\lambda_a}{\lambda_a + \lambda_b}$. Then $\mathbf{y}_1 = \alpha_1 \mathbf{x}_a + (1 - \alpha_1) \mathbf{x}_b$. Take $\beta = \lambda_a + \lambda_b$.
3. While there are elements in the list, set $\alpha_k = \frac{\lambda_j}{\beta + \lambda_j}$, $\mathbf{y}_k = \alpha_k \mathbf{x}_j + (1 - \alpha_k) \mathbf{y}_{k-1}$, and $\beta := \beta + \lambda_j$.
4. Finally, $\mathbf{x} = \mathbf{y}_{p-1}$.

Notice that for each iterate, $\mathbf{y}_k = \alpha_k \mathbf{x}_j + (1 - \alpha_k) \mathbf{y}_{k-1}$, the following inequality follows from convexity of f :

$$f(\mathbf{y}_k) \leq \alpha_k f(\mathbf{x}_j) + (1 - \alpha_k) f(\mathbf{y}_{k-1}) \leq \max\{f(\mathbf{x}_j), f(\mathbf{y}_{k-1})\}.$$

Also for $\mathbf{y}_1 = \alpha_1 \mathbf{x}_a + (1 - \alpha_1) \mathbf{x}_b$

$$f(\mathbf{y}_1) \leq \alpha_1 f(\mathbf{x}_a) + (1 - \alpha_1) f(\mathbf{x}_b) \leq \max\{f(\mathbf{x}_a), f(\mathbf{x}_b)\}.$$

Then, for any iterate \mathbf{y}_k , $k = 1, \dots, p - 1$:

$$f(\mathbf{y}_k) \leq \max\{f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_{n_e})\}.$$

Since this is true for any point $\mathbf{x} \in X$, then, the function f attains a maximum at one of the extreme points of X .

A lower bound can be computed by solving the following LP:

$$\begin{aligned}
\text{LBD} = \min_{\mathbf{v} \in \mathbb{R}^{nv}, \mathbf{b} \in \mathbb{R}^{nm}} \mathbf{c}^T \mathbf{v}, \\
\text{s.t. } \mathbf{A} \mathbf{v} - \mathbf{b} = \mathbf{0}, \\
b_i^L \leq b_i \leq b_i^U, \forall i = 1, \dots, k, \\
\mathbf{v} \geq \mathbf{0}.
\end{aligned} \tag{C.8}$$

C.3 Convex and concave relaxations of compositions of h .

Consider an open set $D_p \subset \mathbb{R}^p$ and a closed convex set $P \subset D_p$ and let $\mathbf{b} : D_p \rightarrow \mathbb{R}^m$. With convex and concave relaxations available for h on B , expressions (C.1) and (C.2) can be used to obtain convex and concave relaxations of $\hat{h} \equiv h \circ \mathbf{b}$ on P . When using (C.8) and (C.7) as convex and concave relaxations of h on B , the evaluation of (C.1) and (C.2) results trivial. However, when h and (C.5) are used as convex and concave relaxations of h on B , the implementation of (C.1) and (C.2) requires the solution of LPs.

Consider convex and concave relaxations $\mathbf{b}^{cv}, \mathbf{b}^{cc}$ for \mathbf{b} on P are available through standard McCormick relaxations. Then for $\mathbf{p} \in P$, $\mathbf{b}^{cv}(\mathbf{p})$ and $\mathbf{b}^{cc}(\mathbf{p})$ define a box $\tilde{B} \subset B$. Let \hat{h}^{cv} and \hat{h}^{cc} be convex and concave relaxations of \hat{h} on P . Then \hat{h}^{cv} is

$$\begin{aligned}
\hat{h}^{cv}(\mathbf{p}) = \min_{\mathbf{v} \in \mathbb{R}^{nv}, \mathbf{b} \in \mathbb{R}^{nm}} \mathbf{c}^T \mathbf{v}, \\
\text{s.t. } \mathbf{A} \mathbf{v} - \mathbf{z} = \mathbf{0}, \\
\mathbf{v} \geq \mathbf{0}, \\
b_i^{cv}(\mathbf{p}) \leq b_i \leq z_i^{cc}(\mathbf{p}), \forall i = 1, \dots, m,
\end{aligned} \tag{C.9}$$

and \widehat{h}^{cc}

$$\begin{aligned}
\widehat{h}^{cc}(\mathbf{p}) = & \max_{\lambda \in \mathbb{R}^{2^k}, \mathbf{z} \in \mathbb{R}^m} \sum_{i=1}^{2^k} \lambda_i h(\mathbf{b}^i) \\
\text{s.t.} & \sum_{i=1}^{2^k} \lambda_i \mathbf{b}^i = \mathbf{z}, \\
& \sum_{i=1}^{2^k} \lambda_i = 1, \\
& b_j^{cv}(\mathbf{p}) \leq z_j \leq b_j^{cc}(\mathbf{p}), \quad \forall j = 1, \dots, m, \\
& \lambda_i \geq 0, \quad \forall i = 1, \dots, 2^k,
\end{aligned} \tag{C.10}$$

$\forall \mathbf{p} \in P$ where all \mathbf{b}^i are computed beforehand and correspond to the 2^k different combinations of the variable components of the right-hand side. The inequalities $b_j^{cv}(\mathbf{p}) \leq b_j \leq b_j^{cc}(\mathbf{p})$ define the box \tilde{B} . These inequalities contain $m - k$ equalities because there are only k variable components of \mathbf{b} . There are 2^k combinations \mathbf{b}^i of lower and upper bounds for the variable components of \mathbf{b} .

Finally, regardless of whether (C.8) and (C.7) or (C.9) and (C.10) are used to compute h^{cv} and h^{cc} , Definition 15 of [121] can be used to generate convex and concave relaxations of $f \circ h \circ \mathbf{b}$ on P for any factorable objective function f .

C.4 Procedure to Calculate Convex and Concave Relaxations of $f \circ h \circ \mathbf{b}$ on P

1. Generate convex and concave relaxations of \mathbf{b} on P using standard McCormick Relaxations.
2. Use the interval bounds for \mathbf{b} on P to compute the 2^k different $h(\mathbf{b}_i)$.
3. Compute (C.8) and (C.7) to obtain lower and upper bounds for $h \circ \mathbf{b}$ on P .
4. Use either (C.8) and (C.7) or (C.9) and (C.10) to compute convex and concave relaxations of \widehat{h} on P .

5. Initialize a McCormick object with lower and upper bounds calculated on step 3 and convex and concave relaxations calculated on step 4.
6. Use Definition 15 on [121] to compute convex and concave relaxations of $f \circ h \circ \mathbf{b}$ on P .

An example implementing this procedure can be found in Section C.6.2.

C.5 Extension to Lexicographic LPs

In DFBA models, the solution set of (6.1) is often nonunique and a lexicographic LP must be solved. Following, convex and concave relaxations of a lexicographic LP will be presented. Consider the case where $n_h > 1$.

Assume that upper and lower bounds, and convex and concave relaxations of h_j for $j \in \{1, \dots, i-1\}$ are available. Consider the function h_i^{LB} :

$$\begin{aligned}
 h_i^{LB}(\mathbf{z}, m_1, n_1, \dots, m_{i-1}, n_{i-1}) = & \\
 \min_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_i^T \mathbf{v}, & \\
 \text{s.t. } \mathbf{A} \mathbf{v} = \mathbf{z}, & \\
 \begin{bmatrix} \mathbf{c}_1^T \\ -\mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_{i-1}^T \\ -\mathbf{c}_{i-1}^T \end{bmatrix} \mathbf{v} \leq \begin{bmatrix} m_1 \\ -n_1 \\ \vdots \\ m_{i-1} \\ -n_{i-1} \end{bmatrix}, & \quad (\text{C.11}) \\
 \mathbf{v} \geq \mathbf{0}. &
 \end{aligned}$$

Notice that if $m_j = h_j(\mathbf{z})$ and $n_j = h_j(\mathbf{z})$, then $h_i^{LB}(\mathbf{z}, m_1, n_1, m_2, n_2, \dots, m_{i-1}, n_{i-1}) = h_i(\mathbf{z})$. If $n_1 \leq h_i(\mathbf{z}) \leq m_1$, then $h_i^{LB}(\mathbf{z}, m_1, n_1, m_2, n_2, \dots, m_{i-1}, n_{i-1}) \leq h_i(\mathbf{z})$. Convex relaxations of h_i^{LB} are also convex relaxations of h_i . An upper bound can be obtained

following the same reasoning:

$$\begin{aligned}
h_i^{UB}(\mathbf{z}, m_1, n_1, \dots, m_{i-1}, n_{i-1}) = & \\
& \max_{\mathbf{v} \in \mathbb{R}^{n_v}} \mathbf{c}_i^T \mathbf{v}, \\
& \text{s.t. } \mathbf{A} \mathbf{v} = \mathbf{z}, \\
& \begin{bmatrix} \mathbf{c}_1^T \\ -\mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_{i-1}^T \\ -\mathbf{c}_{i-1}^T \end{bmatrix} \mathbf{v} \leq \begin{bmatrix} m_1 \\ -n_1 \\ \vdots \\ m_{i-1} \\ -n_{i-1} \end{bmatrix}, \\
& \mathbf{v} \geq \mathbf{0}.
\end{aligned} \tag{C.12}$$

Then, any concave relaxations of h_i^{UB} will be concave relaxations of $h_i(\mathbf{z})$. Following is a proof that h_i^{LB} or h_i^{UB} cannot be unbounded. Assume that $\mathbf{z} \in F$. Then, if $m_j = h_j(\mathbf{z})$ and $n_j = h_j(\mathbf{z})$ for $j \in \{1, \dots, n_h\}$, (C.11) has a solution. Then, from strong duality the dual program of (C.11) shown in (C.14) has a solution equal to (C.11). Following is (C.11) in standard form:

$$\begin{aligned}
h_i^{LB}(\mathbf{z}, m_1, n_1, \dots, m_{i-1}, n_{i-1}) = & \min_{\mathbf{v} \in \mathbb{R}^{n_v+2i}} \widehat{\mathbf{c}}_i^T \mathbf{v}, \\
& \text{s.t. } \begin{bmatrix} \mathbf{A} & \mathbf{Z} \\ \mathbf{C} & \mathbf{I}_{2i} \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{z} \\ \mathbf{m} \end{bmatrix}, \\
& \mathbf{v} \geq \mathbf{0},
\end{aligned} \tag{C.13}$$

where $\mathbf{Z} \in \mathbb{R}^{m \times 2i}$ is a matrix made of zeros, $\mathbf{C} \in \mathbb{R}^{2i \times n_v}$ contains the cost vectors in (C.11), $\widehat{\mathbf{c}}_i^T = \begin{bmatrix} \mathbf{c}_i^T & \mathbf{0}^T \end{bmatrix}$ with $\mathbf{0} \in \mathbb{R}^{2i}$ and $\mathbf{m} \in \mathbb{R}^{2i}$ contains the m_j and n_j entries in (C.11). Then, the dual program of (C.11) is:

$$\begin{aligned}
& \max_{\boldsymbol{\lambda} \in \mathbb{R}^{n_m+2i}} \boldsymbol{\lambda}^T \begin{bmatrix} \mathbf{z} \\ \mathbf{m} \end{bmatrix}, \\
& \text{s.t.} \quad \begin{bmatrix} \mathbf{A}^T & \mathbf{C}^T \\ \mathbf{Z}^T & \mathbf{I}_{2i} \end{bmatrix} \leq \widehat{\mathbf{c}}_i.
\end{aligned} \tag{C.14}$$

Notice that if (C.14) has a solution, for changes in \mathbf{z} or \mathbf{m} , (C.14) remains feasible. The dual program of (C.12) is the same as (C.14) but with $-\widehat{\mathbf{c}}_i$ instead of $\widehat{\mathbf{c}}_i$. Since $\boldsymbol{\lambda}$ is free, if (C.14) has a feasible point, then the negative of this point is feasible in the dual of (C.12). Then, (C.11) and (C.12) cannot be unbounded. In addition, (C.11) and (C.12) will be feasible if $h_i(\mathbf{z}) \in [n_i, m_i]$ for all $i = 1, \dots, n_h$. Let n_i, m_i take the values of the lower bounds and upper bounds respectively of h_i for all $i = 1, \dots, n_h$. Notice that (C.11) is a convex function and (C.12) is a concave function with respect to changes on the right-hand side, then they are convex and concave relaxations of (7.3).

Lower and upper bounds can be computed using a similar version of (C.8) on (C.11) and (C.12) by adding variable right-hand sides in \mathbf{z} as variables in the LP with inequalities constraining their values. Convex and concave relaxations of $\widehat{\mathbf{h}} \equiv \mathbf{h} \circ \mathbf{b}$ on P can be obtained using composition Theorem C.1.4. For $i = 2, \dots, n_h$:

$$\begin{aligned}
\widehat{h}_i^{cv}(\mathbf{p}) &= \min_{\mathbf{v} \in \mathbb{R}^{n_v}, \mathbf{b} \in \mathbb{R}^m} \mathbf{c}_i^T \mathbf{v}, \\
& \text{s.t.} \quad \mathbf{A} \mathbf{v} - \mathbf{b}(\mathbf{p}) = \mathbf{0}, \\
& \begin{bmatrix} \mathbf{c}_1^T \\ -\mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_{i-1}^T \\ -\mathbf{c}_{i-1}^T \end{bmatrix} \mathbf{v} \leq \begin{bmatrix} UBD_1 \\ -LBD_1 \\ \vdots \\ UBD_{i-1} \\ -LBD_{i-1} \end{bmatrix}, \\
& \mathbf{v} \geq \mathbf{0}, \quad b_i^{cv}(\mathbf{p}) \leq b_i \leq b_i^{cc}(\mathbf{p}), \quad \forall i = 1, \dots, m,
\end{aligned} \tag{C.15}$$

where LBD_i, UBD_i are the lower and upper bounds of h_i on P . As before, most elements of \mathbf{b} will be fixed, therefore, the last set of inequalities in (C.15) are equalities for most components of \mathbf{b} . Similarly,

$$\begin{aligned} \widehat{h}_i^{cc}(\mathbf{p}) = & \max_{\mathbf{v} \in \mathbb{R}^{n_v}, \mathbf{b} \in \mathbb{R}^n} \mathbf{c}_i^T \mathbf{v}, \\ & \text{s.t. } \mathbf{A}\mathbf{v} - \mathbf{b}(\mathbf{p}) = \mathbf{0}, \\ & \begin{bmatrix} \mathbf{c}_1^T \\ -\mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_{i-1}^T \\ -\mathbf{c}_{i-1}^T \end{bmatrix} \mathbf{v} \leq \begin{bmatrix} UBD_1 \\ -LBD_1 \\ \vdots \\ UBD_{i-1} \\ -LBD_{i-1} \end{bmatrix}, \\ & \mathbf{v} \geq \mathbf{0}, \quad b_i^{cv}(\mathbf{p}) \leq b_i \leq b_i^{cc}(\mathbf{p}), \quad \forall i = 1, \dots, n_m. \end{aligned} \tag{C.16}$$

For $i = 1$, (C.9) and (C.10) are used to obtain convex and concave relaxations of $h_i \circ \mathbf{b}$ on P .

C.6 Examples

C.6.1 Concave envelope of an LP with respect to its right-hand side

Consider $b_1 \in [7, 11]$, $b_2 \in [1000, 1400]$ and the following LP:

$$\begin{aligned} h(b_1, b_2) = & \min_{\mathbf{x} \in \mathbb{R}^2} -500x_1 - 300x_2, \\ & \text{s.t. } x_1 + x_2 \leq b_1, \\ & x_1 + x_2 \geq 7, \\ & 200x_1 + 100x_2 \leq b_2 \\ & x_1 + 2x_2 \leq 12 \\ & x_1, x_2 \geq 0. \end{aligned} \tag{C.17}$$

Since h is a convex function with respect to \mathbf{b} , no convex relaxation needs to be calculated. Figure C-1, shows the concave envelope of (C.17) on $[7, 11] \times [1000, 1400]$. Image D in Figure C-1 shows how smaller intervals yield better concave relaxations. In particular, the concave envelope on $[9, 11] \times [1000, 1400]$ coincides with the original function.

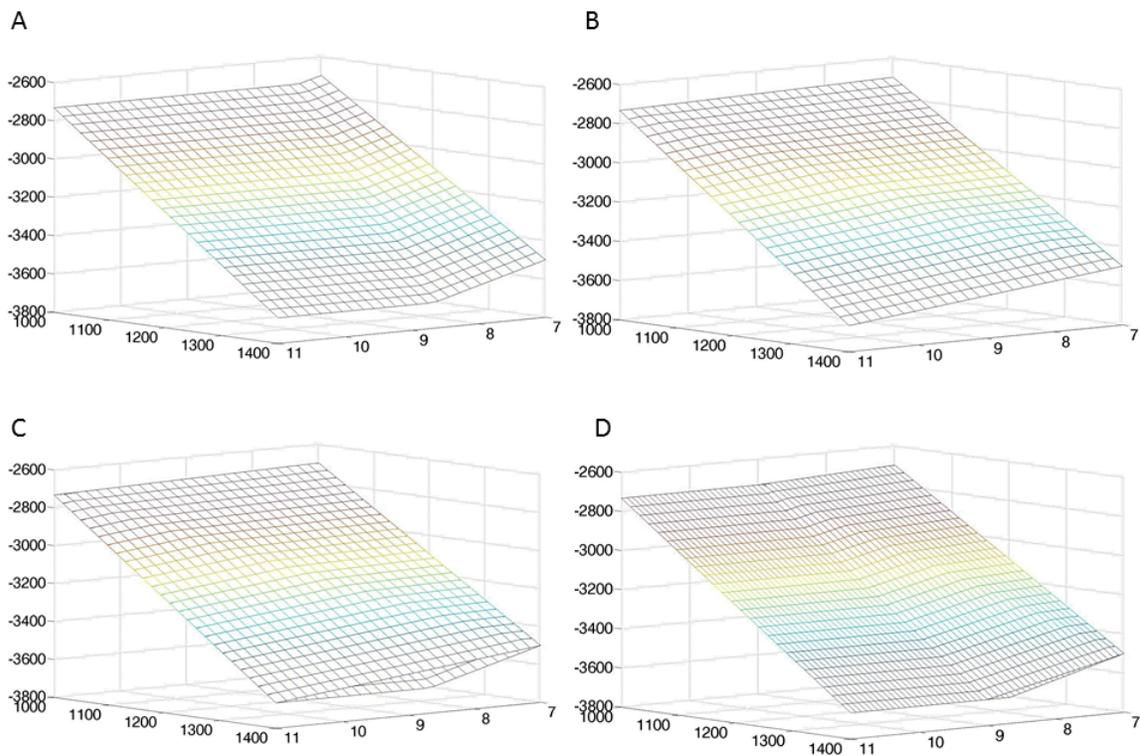


Figure C-1: Convex and concave envelopes for function (C.17). A) Original function on $[7, 11] \times [1000, 1400]$. B) Concave envelope of (C.17) on $[7, 11] \times [1000, 1400]$ using (C.5). C) Original function on $[7, 11] \times [1000, 1400]$ and concave envelope on $[7, 11] \times [1000, 1400]$. D) Original function on $[7, 11] \times [1000, 1400]$ and concave envelopes on $[7, 9] \times [1000, 1400]$, and $[9, 11] \times [1000, 1400]$.

C.6.2 Convex and concave relaxations of factorable functions with an LP embedded

Consider $p_1 \in [0, 2]$, $p_2 \in [0, 2]$, and the following LP:

$$\begin{aligned} h(p_1, p_2) &= \min_{\mathbf{v} \in \mathbb{R}^3} 2v_1 - v_2, \\ \text{s.t. } v_1 + v_2 &\leq p_1^2, \\ v_1 - v_3 &\leq p_2, \\ v_2 + v_3 &\leq p_1 + p_2 \\ v_1, v_2, v_3 &\geq 0. \end{aligned} \tag{C.18}$$

Finally, consider the factorable function:

$$g = 0.5h^3 + 0.005h^2 - 10h. \tag{C.19}$$

Convex and concave relaxations of h on $[0, 2] \times [0, 2]$ can be seen in Figure (C-2). Convex and concave relaxations of g on $[0, 2] \times [0, 2]$ can be seen in Figure C-3. When upper and lower bounds are used as convex and concave relaxations, weaker relaxations are obtained as seen in Figure C-4. Finally, the relaxations approximate better the function in smaller intervals as seen in Figure C-5.

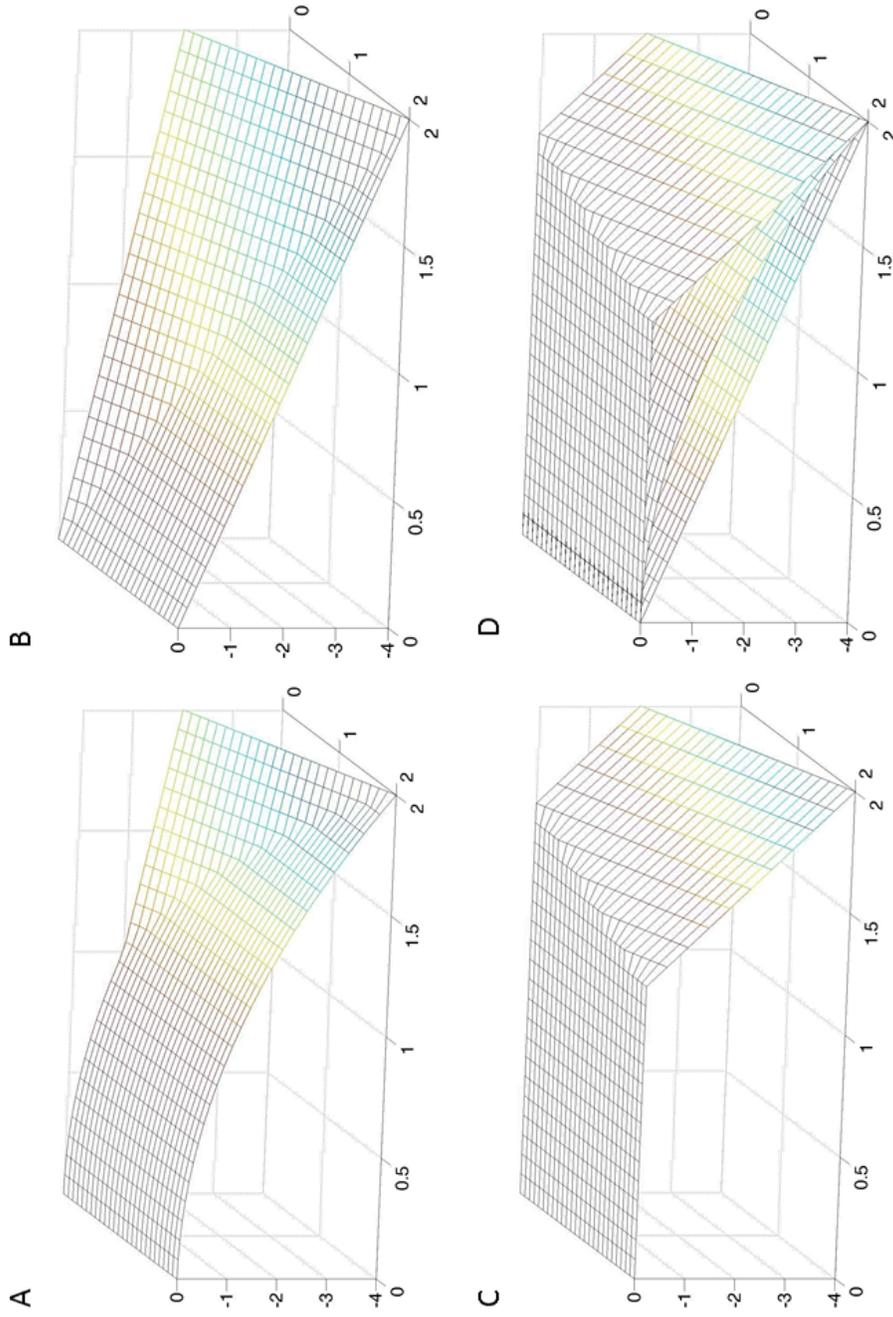


Figure C-2: Convex and concave relaxations of (C.18) on $[0, 2] \times [0, 2]$ calculated using (C.9) and (C.10). A) Original function. B) Convex relaxation. C) Concave relaxation. D) Original function with convex and concave relaxations.

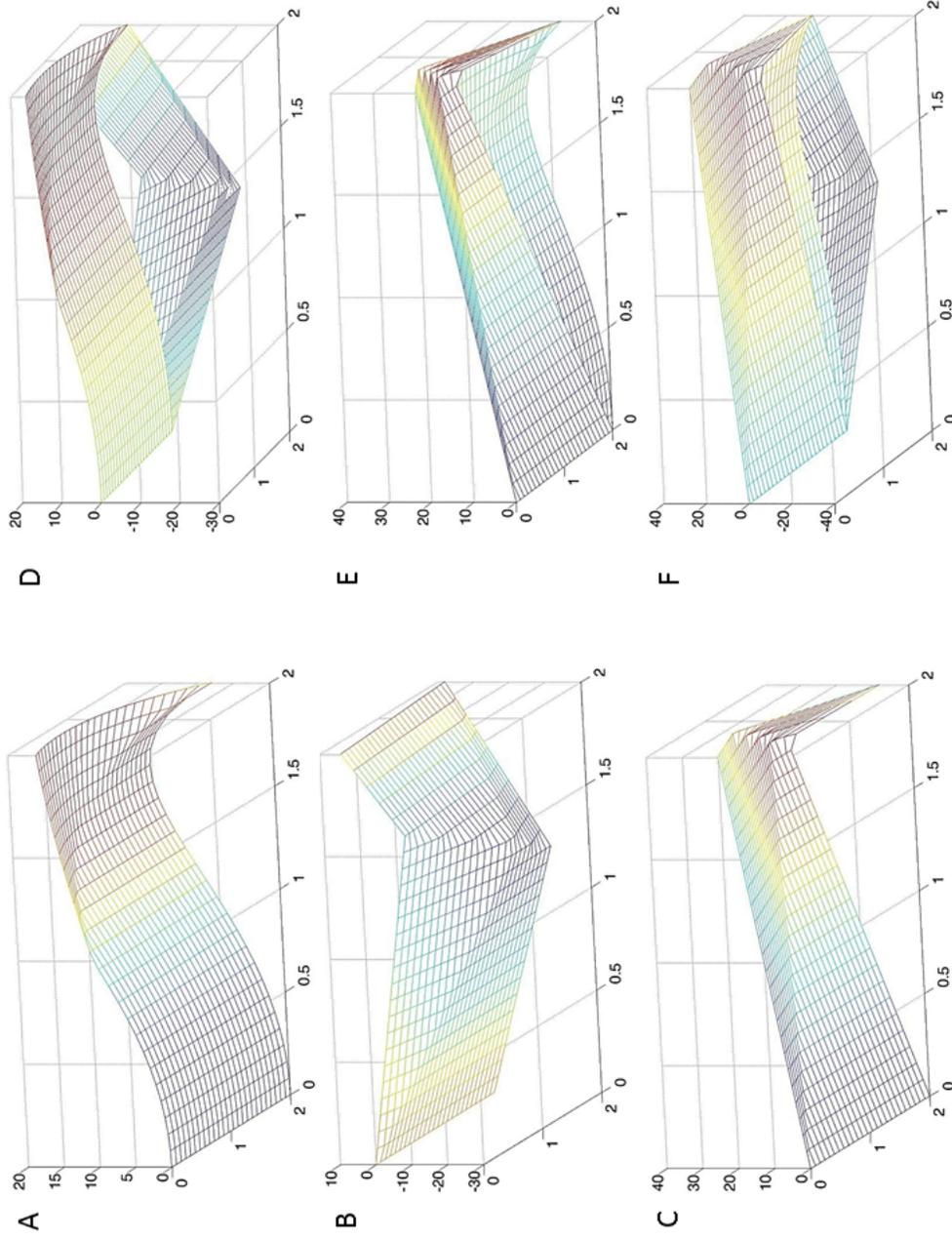


Figure C-3: Convex and concave relaxations of (C.19) on $[0, 2] \times [0, 2]$ calculated using (C.9) and (C.10). A) Original function. B) Concave relaxation. C) Convex relaxation. D) Original function with concave relaxation. E) Original function with convex relaxation. F) Original function with convex and concave relaxations.

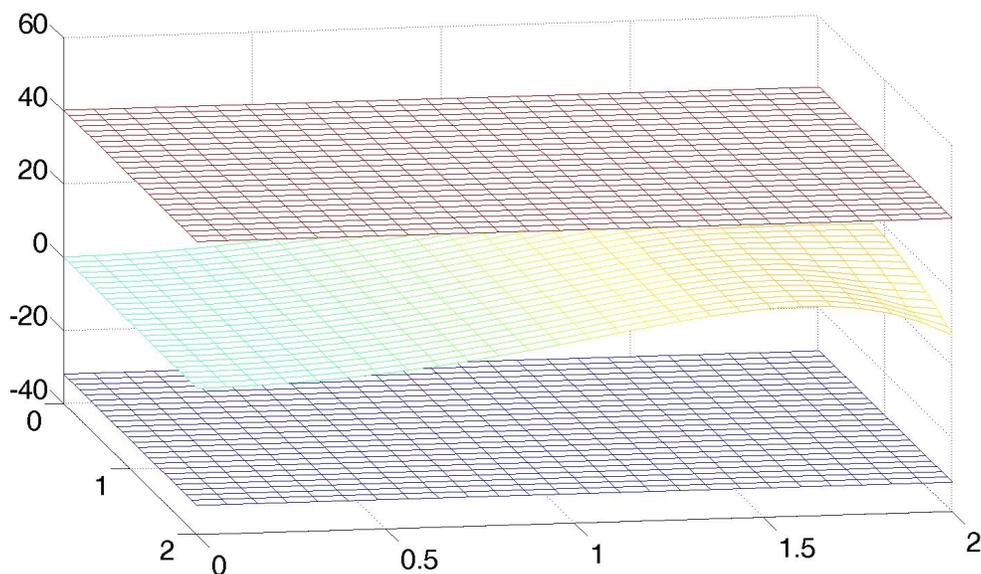


Figure C-4: Upper and lower bounds of (C.19) on $[0, 2] \times [0, 2]$ calculated using (C.8) and (C.7).

C.6.3 Convex and Concave Relaxations of factorable functions with a Lexicographic LP embedded

Consider an *E. coli* culture with glucose (p_1) and xylose (p_2) concentrations as variables. Assume that the glucose and xylose uptakes can be bounded by the following Michaelis-Menten expressions:

$$v_{glucose}^{UB} = 10.5 \left(\frac{p_1}{0.0027 + p_1} \right),$$

$$v_{xylose}^{UB} = 6 \left(\frac{p_2}{0.0165 + p_2} \right) \left(\frac{1}{1 + \frac{p_1}{5}} \right).$$

The parameters were obtained from [51] and slightly modified to illustrate the performance of the relaxations. The *E. coli* model used was iJR904 from [107] and considers 761 metabolites and 1075 reactions. The lexicographic linear program had the following objectives: maximize biomass (h_1), maximize glucose consumption (h_2), and maximize xylose consumption (h_3), respectively. Then, the following objective

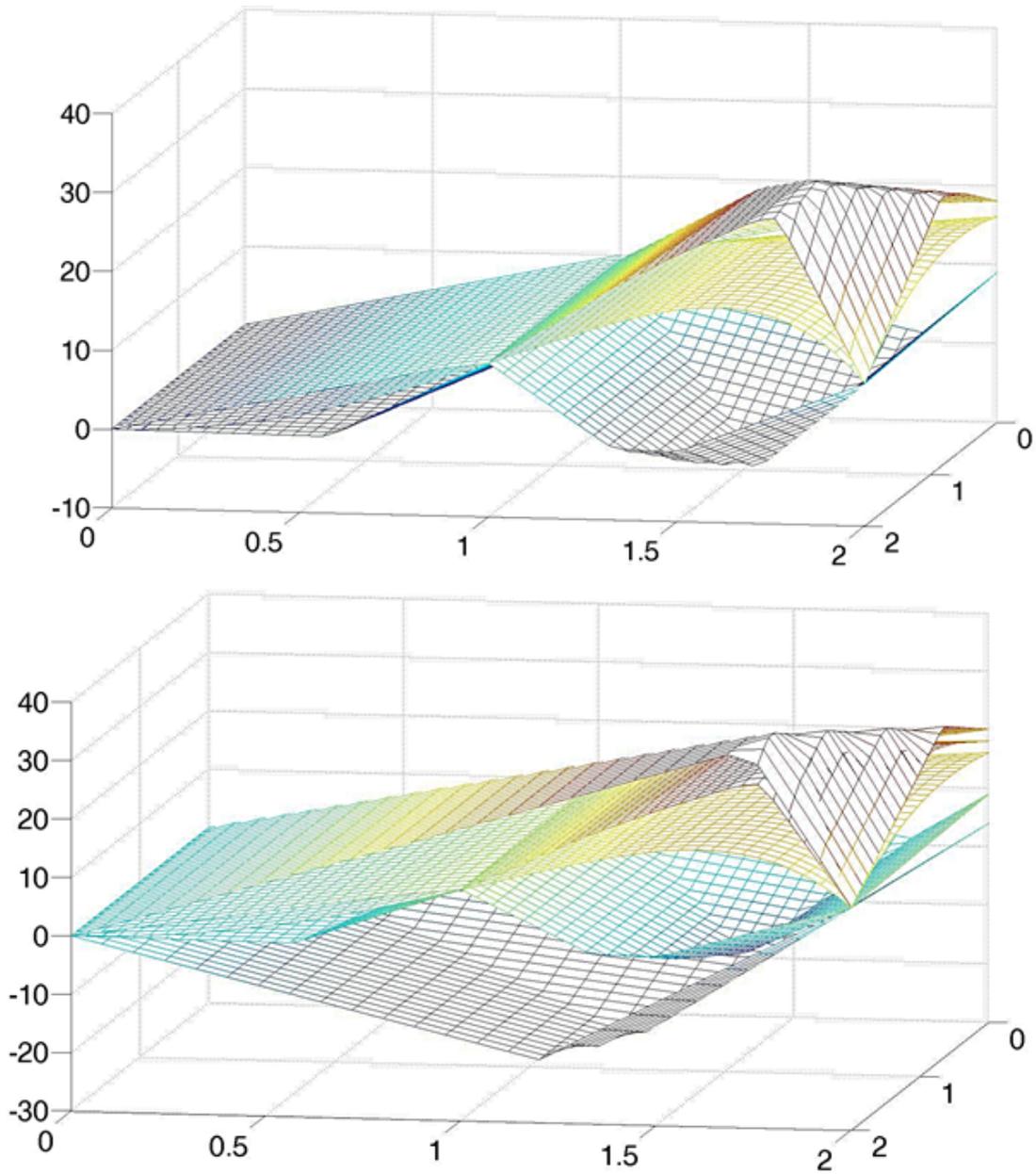


Figure C-5: Convex and concave relaxations of (C.19) calculated using (C.9) and (C.10) on smaller sections. Top: Original function with convex and concave relaxations on $[0, 1] \times [0, 2]$ and on $[1, 2] \times [0, 2]$. Bottom: Original function with convex and concave relaxations on $[0, 1] \times [0, 2]$, $[1, 2] \times [0, 2]$, and on $[0, 2] \times [0, 2]$.

function dependent on the objective function values of the lexicographic LP was formulated:

$$f(\mathbf{h}) = 100h_1 + 10h_2^2 + 10(-h_3)^3.$$

Concentrations (mmol/L) of glucose were in $[0, 0.00025]$ and of xylose in $[10, 60]$. Figures C-6, C-7, and C-8 show the convex and concave relaxations obtained for $\mathbf{h} \circ \mathbf{b}$ on $[0, 0.00025] \times [10, 60]$ and for $f \circ \mathbf{h} \circ \mathbf{b}$ on $[0, 0.00025] \times [10, 60]$. It can be seen that tight convex relaxations and weak concave relaxations for $\mathbf{h} \circ \mathbf{b}$ on $[0, 0.00025] \times [10, 60]$ are obtained. The weak concave relaxations result in weak convex and concave relaxations for $f \circ \mathbf{h} \circ \mathbf{b}$ on $[0, 0.00025] \times [10, 60]$.

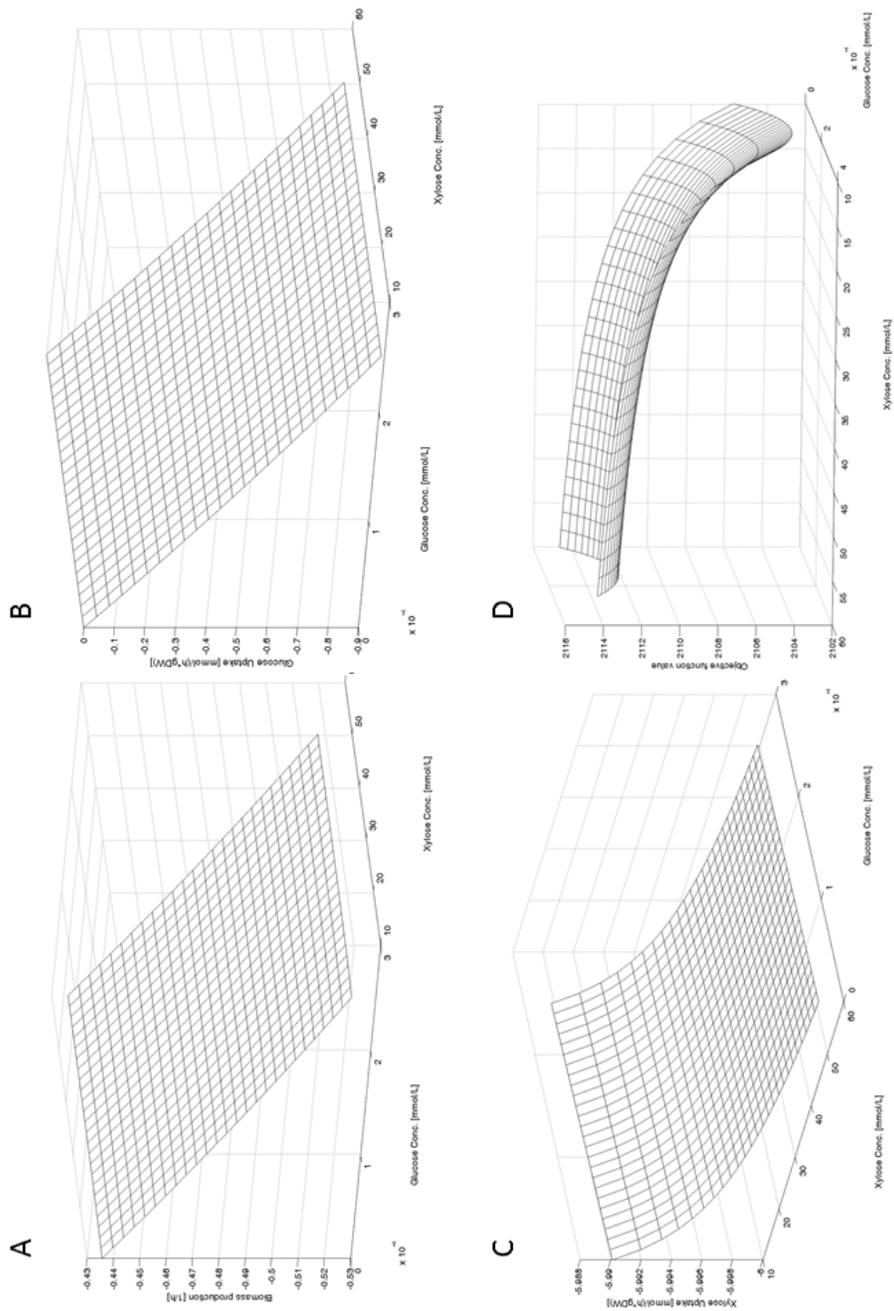


Figure C-6: Plots of $\mathbf{h} \circ \mathbf{b}$ and $f \circ \mathbf{h} \circ \mathbf{b}$ for $p_1 \in [0, 0.00025]$ and $p_2 \in [10, 60]$. A) Biomass production rate (1/h). B) Glucose consumption rate (mmol/(h*gDW)). C) Xylose consumption rate (mmol/(h*gDW)). D) Plot of $f \circ \mathbf{h} \circ \mathbf{b}$ on $[0, 0.00025] \times [10, 60]$.

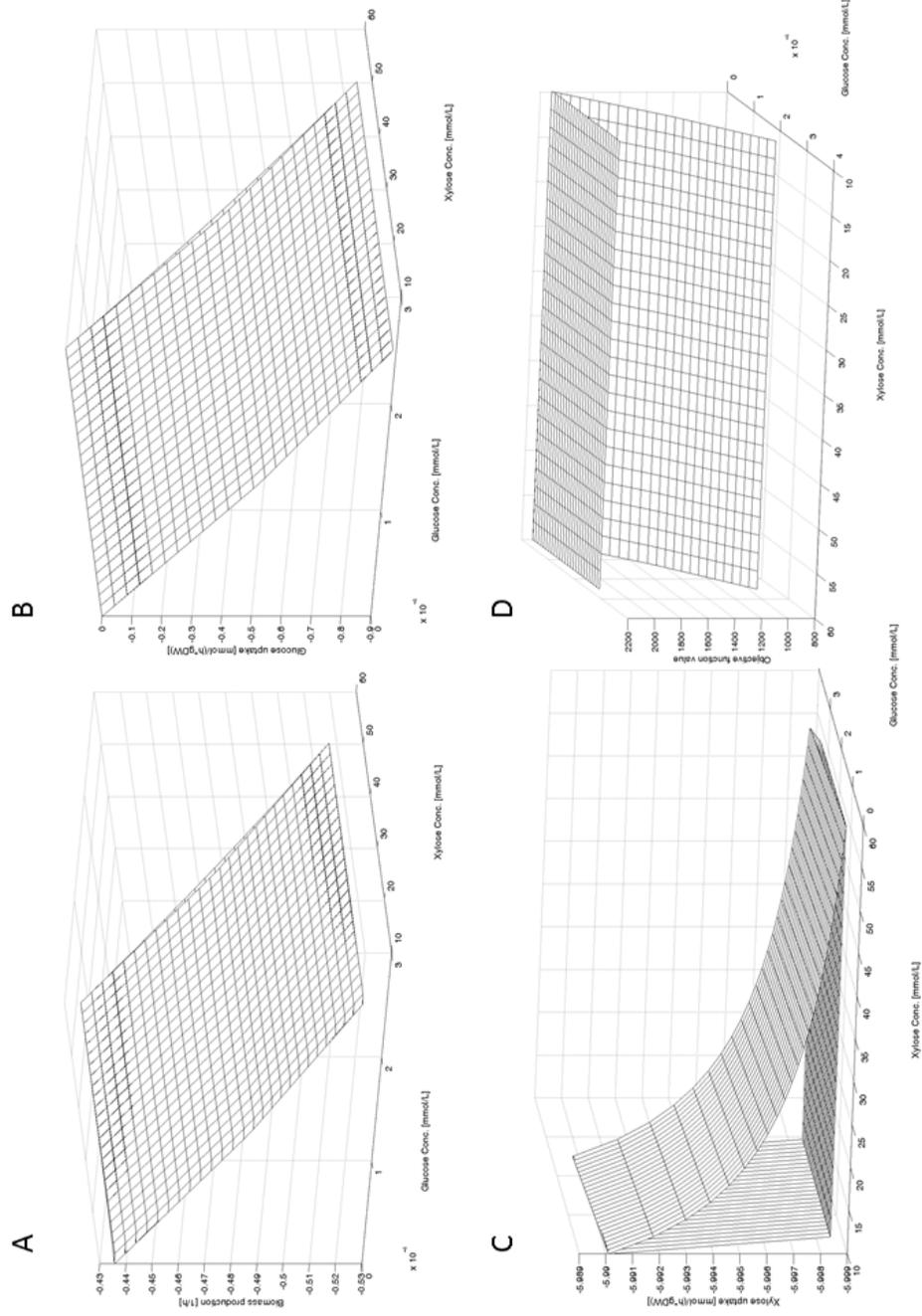


Figure C-7: Plots of $\mathbf{h} \circ \mathbf{b}$ and $f \circ \mathbf{h} \circ \mathbf{b}$ with convex relaxations for $p_1 \in [0, 0.00025]$ and $p_2 \in [10, 60]$. A) Biomass production rate ($1/h$). B) Glucose consumption rate ($\text{mmol}/(\text{h} \cdot \text{gDW})$). C) Xylose consumption rate ($\text{mmol}/(\text{h} \cdot \text{gDW})$). D) Plot of $f \circ \mathbf{h} \circ \mathbf{b}$ on $[0, 0.00025] \times [10, 60]$.

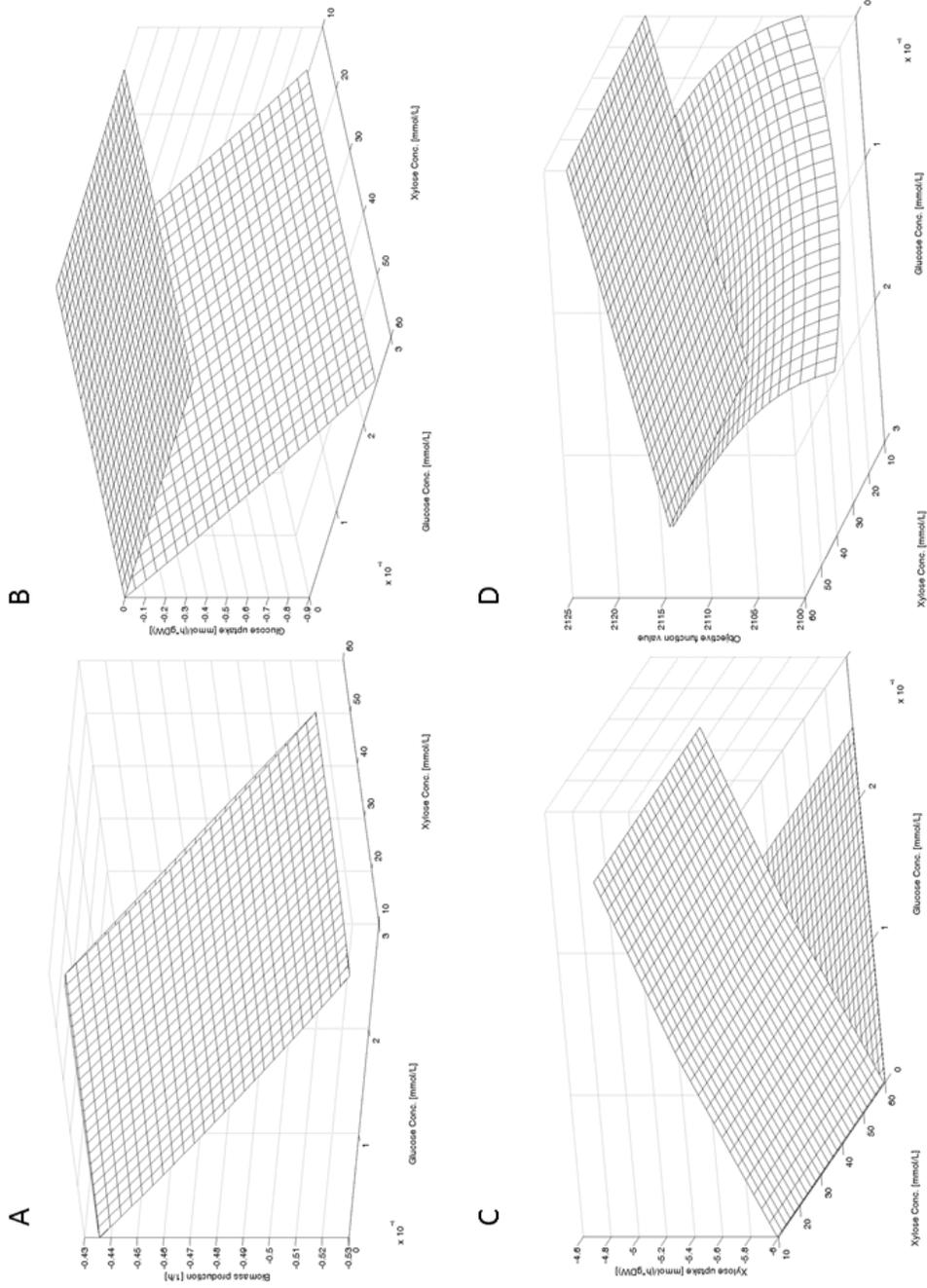


Figure C-8: Plots of $\mathbf{h} \circ \mathbf{b}$ and $f \circ \mathbf{h} \circ \mathbf{b}$ with concave relaxations for $p_1 \in [0, 0.00025]$ and $p_2 \in [10, 60]$. A) Biomass production rate (1/h). B) Glucose consumption rate (mmol/(h*gDW)). C) Xylose consumption rate (mmol/(h*gDW)). D) Plot of $f \circ \mathbf{h} \circ \mathbf{b}$ on $[0, 0.00025] \times [10, 60]$.

C.7 Conclusions

A method to calculate convex and concave relaxations of factorable functions with an LP embedded has been presented. This method uses generalized McCormick relaxations, the multivariate McCormick composition Theorem, and the concave envelope of an LP with respect to its right-hand side to compute these relaxations. Evaluating these relaxations requires the solution of several LPs. Upper and lower bounds are cheaper to compute, but convex and concave relaxations equal to these bounds are much weaker. This method can be implemented in problems with LPs embedded where the number of variable right-hand sides k is relatively small. In a branch and bound algorithm, initially 2^k LPs need to be solved to construct the concave relaxation h^{cc} in (C.10). Then, everytime the domain is branched, 2^{k-1} new LPs need to be solved to construct h^{cc} for each node. Convergence properties of the relaxations to the original function on domains that tend to zero remain to be proven.

Convex and concave relaxations for factorable functions with lexicographic linear programs embedded were obtained. The cost of computing these relaxations increases linearly with the number of levels of optimization (n_h). Tight convex relaxations and weak concave relaxations were obtained for $\mathbf{h} \circ \mathbf{b}$ on P . Better concave relaxations are needed to obtain tighter convex and concave relaxations of $f \circ \mathbf{h} \circ \mathbf{b}$ on P .

Bibliography

- [1] Ilan Adler and Renato DC Monteiro. A geometric view of parametric linear programming. *Algorithmica*, 8(1):161–176, 1992.
- [2] Rakesh Agrawal and Navneet R Singh. Solar energy to biofuels. *Annual review of chemical and biomolecular engineering*, 1:343–364, 2010.
- [3] Rakesh Agrawal, Navneet R Singh, Fabio H Ribeiro, and W Nicholas Delgass. Sustainable fuel for the transportation sector. *Proceedings of the National Academy of Sciences*, 104(12):4828–4833, 2007.
- [4] Luke Amer, Birendra Adhikari, and John Pellegrino. Technoeconomic analysis of five microalgae-to-biofuels processes of varying complexity. *Bioresource Technology*, 102(20):9350–9359, 2011.
- [5] Najmul Arifeen, Ruohang Wang, Ioannis Kookos, Colin Webb, and Apostolis A Koutinas. Optimization and cost estimation of novel wheat biorefining for continuous production of fermentation feedstock. *Biotechnology progress*, 23(4):872–880, 2007.
- [6] Konstantine D. Balkos and Brian Colman. Mechanism of CO₂ acquisition in an acid-tolerant *Chlamydomonas*. *Plant, Cell and Environment*, 30:745–752, 2007.
- [7] Paul I. Barton and Jose A. Gomez. Dynamic flux balance analysis using DF-BAlab. In Marco Fondi, editor, *Metabolic Network Reconstruction and Modeling: Methods and Protocols*. Springer Science+Business Media, In Press.
- [8] Paul I. Barton, Jose A. Gomez, and Kai Höffner. Production of biofuels from sunlight and lignocellulosic sugars using microbial consortia. In *Proceedings of the 2nd Southeast European Conference on Sustainable Development of Energy, Water and Environment Systems*, 2016.
- [9] Colin M Beal, Léda N Gerber, Deborah L Sills, Mark E Huntley, Stephen C Machesky, Michael J Walsh, Jefferson W Tester, Ian Archibald, Joe Granados, and Charles H Greene. Algal biofuel production for fuels and feed in a 100-ha facility: A comprehensive techno-economic analysis and life cycle assessment. *Algal Research*, 10:266–279, 2015.
- [10] EW Becker. *Microalgae: Biotechnology and Bioengineering*. Cambridge Univ Press, 1994.

- [11] Athanasios Beopoulos, Julien Cescut, Ramdane Haddouche, Jean-Louis Uribe-larrea, Carole Molina-Jouve, and Jean-Marc Nicaud. *Yarrowia lipolytica* as a model for bio-oil production. *Progress in Lipid Research*, 48(6):375–387, 2009.
- [12] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997.
- [13] Nadine C Boelee, Hardy Temmink, Marcel Janssen, Cees JN Buisman, and René H Wijffels. Scenario analysis of nutrient removal from municipal wastewater by microalgal biofilms. *Water*, 4(2):460–473, 2012.
- [14] J Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, New York, 2013.
- [15] S. Boyd and L. Vandenberghe. Subgradients: Notes for EE364b, Stanford University, Winter 2006-07. https://see.stanford.edu/materials/lsoctee364b/01-subgradients_notes.pdf, 2008.
- [16] Guido Breuer, Packo P Lamers, Dirk E Martens, René B Draaisma, and René H Wijffels. The impact of nitrogen starvation on the dynamics of triacylglycerol accumulation in nine microalgae strains. *Bioresource Technology*, 124:217–226, 2012.
- [17] Lewis M Brown. Uptake of carbon dioxide from flue gas by microalgae. *Energy Conversion and Management*, 37(6):1363–1367, 1996.
- [18] H. O. Buhr and S. B. Miller. A dynamic model of the high-rate algal-bacterial wastewater treatment pond. *Water Res.*, 17:29–37, 1983.
- [19] Roger L. Chang, Lila Ghamsari, Ani Manichaikul, Erik F.Y. Hom, Santhanam Balaji, Weiqi Fu, Yun Shen, Tong Hao, Bernhard Ø. Palsson, Kourosch Salehi-Ashtiani, and Jason A. Papin. Metabolic network reconstruction of *Chlamydomonas* offers insight into light-driven algal metabolism. *Molecular Systems Biology*, 7(518), 2011.
- [20] Charles River Watershed Association, 190 Park Rd, Weston, MA 02453. *Total Maximum Daily Load for Nutrients in the Upper/Middle Charles River, Massachusetts*, 2011.
- [21] Benjamas Cheirsilp, Warangkana Suwannarat, and Rujira Niyomdecha. Mixed culture of oleaginous yeast *Rhodotorula glutinis* and microalga *Chlorella vulgaris* for lipid production from industrial wastes and its use as biodiesel feedstock. *New Biotechnology*, 28(4):362–368, 2011.
- [22] Jin Chen, Jose A Gomez, Kai Höffner, Paul I Barton, and Michael A Henson. Metabolic modeling of synthesis gas fermentation in bubble column reactors. *Biotechnology for biofuels*, 8(89), 2015.

- [23] Jin Chen, Jose A Gomez, Kai Höffner, Poonam Phalak, Paul I Barton, and Michael A Henson. Spatiotemporal modeling of microbial metabolism. *BMC systems biology*, 10(21), 2016.
- [24] Yusuf Chisti. *Microalgal Biotechnology: Potential and Production*, chapter Raceways-based production of algal crude oil. De Gruyter, 2012.
- [25] Yusuf Chisti. Constraints to commercialization of algal fuels. *Journal of Biotechnology*, 167(3):201–214, 2013.
- [26] Andres Clarens and Lisa Colosi. Putting algae’s promise into perspective. *Biofuels*, 1(6):805–808, 2010.
- [27] Frank H Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.
- [28] CPLEX IBM ILOG. 12.7 User’s Manual. https://www.ibm.com/support/knowledgecenter/SSSA5P_12.7.0/ilog.odms.studio.help/pdf/usrcplex.pdf, 2017.
- [29] Ryan Davis, Andy Aden, and Philip T Pienkos. Techno-economic analysis of autotrophic microalgae for fuel production. *Applied Energy*, 88(10):3524–3531, 2011.
- [30] MR Droop. Vitamin B12 and marine ecology. IV. The kinetics of uptake, growth and inhibition in *Monochrysis lutheri*. *Journal of the Marine Biological Association of the United Kingdom*, 48(03):689–733, 1968.
- [31] Natalie C. Duarte, Markus J. Herrgård, and Bernhard Ø. Palsson. Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome-scale metabolic model. *Genome Research*, 14(7):1298–1309, 2004.
- [32] Expression Technologies Inc. Bacterial *E. coli* growth media. <http://www.exptec.com/Expression%20Technologies/Bacteria%20growth%20media.htm>, 2003. Accessed: 2016-12-26.
- [33] Francisco Facchinei, Andreas Fischer, and Markus Herrich. An LP-Newton method: nonsmooth equations, KKT systems, and nonisolated solutions. *Mathematical Programming*, 146(1-2):1–36, 2014.
- [34] Michael C. Ferris and Jong-Shi Pang. Engineering and economic applications of complementarity problems. *SIAM Review*, 39(4):669–713, 1997.
- [35] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Kluwer Academic Publishers, 1988.
- [36] Robert Flassig, Melanie Fachet, Kai Höffner, Paul I Barton, and Kai Sundmacher. Dynamic flux balance modeling to increase the production of high-value compounds in green microalgae. *Biotechnology for Biofuels*, 9(165):1–12, 2016.

- [37] Christodoulos A. Floudas. *Deterministic global optimization: theory, methods, and applications*. Kluwer Academic Publishers, 2000.
- [38] Kevin J Flynn. Modelling multi-nutrient interactions in phytoplankton; balancing simplicity and realism. *Progress in Oceanography*, 56(2):249–279, 2003.
- [39] Michael J Follows, Stephanie Dutkiewicz, Scott Grant, and Sallie W Chisholm. Emergent biogeography of microbial communities in a model ocean. *Science*, 315(5820):1843–1846, 2007.
- [40] Marco Fondi and Pietro Liò. Genome-scale metabolic network reconstruction. *Bacterial Pangenomics: Methods and Protocols*, pages 233–256, 2015.
- [41] Antonio R Franco, Jacobo Cárdenas, and Emilio Fernández. A mutant of *Chlamydomonas reinhardtii* altered in the transport of ammonium and methylammonium. *Molecular and General Genetics MGG*, 206(3):414–418, 1987.
- [42] Aurora Galván, Alberto Quesada, and Emilio Fernández. Nitrate and nitrite are transported by different specific transport systems and by a bispecific transporter in *Chlamydomonas reinhardtii*. *Journal of Biological Chemistry*, 271(4):2088–2092, 1996.
- [43] Cristiana Gomes de Oliveira Dal’Molin, Lake-Ee Quek, Robin W. Palfreyman, and Lars K. Nielsen. AlgaGEM - a genome-scale metabolic reconstruction of algae based on the *Chlamydomonas reinhardtii* genome. *BMC Genomics*, 12:(Suppl 4):S5, 2011.
- [44] Jose A. Gomez, Kai Höffner, and Paul I. Barton. DFBAlab: a fast and reliable matlab code for dynamic flux balance analysis. *BMC Bioinformatics*, 15(1):409, 2014.
- [45] Jose A. Gomez, Kai Höffner, and Paul I. Barton. From sugars to biodiesel using microalgae and yeast. *Green Chemistry*, 18(2):461–475, 2016.
- [46] Jose A. Gomez, Kai Höffner, and Paul I. Barton. Mathematical modeling of a raceway pond system for biofuels production. In *Proceedings of the 26th European Symposium on Computer Aided Process Engineering - ESCAPE 26*, 2016.
- [47] Jose A. Gomez, Kai Höffner, Kamil A. Khan, and Paul I. Barton. Generalized derivatives of lexicographic linear programs. *Submitted to Journal of Optimization Theory and Applications*.
- [48] Harvey J Greenberg. An analysis of degeneracy. *Naval Research Logistics (NRL)*, 33(4):635–655, 1986.
- [49] Harvey J Greenberg. The use of the optimal partition in a linear programming solution for postoptimal analysis. *Operations Research Letters*, 15(4):179–185, 1994.

- [50] Gurobi Optimization. Gurobi optimizer reference manual. URL: <http://www.gurobi.com/documentation/>, 2017.
- [51] Timothy J. Hanly and Michael A. Henson. Dynamic flux balance modeling of microbial co-cultures for efficient batch fermentation of glucose and xylose mixtures. *Biotechnology and Bioengineering*, 108(2):376–385, 2011.
- [52] John DI Harper. *Chlamydomonas* cell cycle mutants. *International Review of Cytology*, 189:131–176, 1999.
- [53] Roger Harrabin. US makes climate pledge to UN. <http://www.bbc.com/news/science-environment-32136006>, 2015.
- [54] Elizabeth H Harris. *The Chlamydomonas sourcebook: introduction to Chlamydomonas and its laboratory use*, volume 1. Academic Press, 2009.
- [55] John L Harwood and Irina A Guschina. The versatility of algae and their lipid metabolism. *Biochimie*, 91(6):679–684, 2009.
- [56] Stuart M. Harwood, Kai Höffner, and Paul I. Barton. Efficient solution of ordinary differential equations with a parametric lexicographic linear program embedded. *Numerische Mathematik*, 133(4):623–653, 2016.
- [57] Mette Hein, M Folager Pedersen, and Kaj Sand-Jensen. Size-dependent nitrogen uptake in micro-and macroalgae. *Marine ecology progress series. Oldendorf*, 118(1):247–253, 1995.
- [58] Kai Höffner and Paul I Barton. Design of microbial consortia for industrial biotechnology. *Computer Aided Chemical Engineering*, 34:65–74, 2014.
- [59] Kai Höffner, Stuart M. Harwood, and Paul I. Barton. A reliable simulator for dynamic flux balance analysis. *Biotechnology and Bioengineering*, 110(3):792–802, 2013.
- [60] Kai Höffner, Kamil A. Khan, and Paul I. Barton. Generalized derivatives of dynamic systems with a linear program embedded. *Automatica*, 63:198–208, 2016.
- [61] Michael Hucka, Andrew Finney, Herbert M Sauro, Hamid Bolouri, John C Doyle, Hiroaki Kitano, Adam P Arkin, Benjamin J Bornstein, Dennis Bray, Athel Cornish-Bowden, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.
- [62] ICIS. Indicative Chemical Prices A-Z. <http://www.icis.com/chemicals/channel-info-chemicals-a-z/>, 2008.
- [63] James P. Ignizio. *Linear programming in single & multiple-objective systems*. Prentice Hall, Englewoods Cliffs, N.J. 07632, 1982.

- [64] Intergovernmental Panel on Climate Change. Summary for policymakers. https://www.ipcc.ch/pdf/assessment-report/ar5/wg3/drafts/fgd/ipcc_wg3_ar5_summary-for-policymakers_approved.pdf, 2014.
- [65] Stanislav Janda, Arnošt Kotyk, and Růžena Tauchová. Monosaccharide transport systems in the yeast *Rhodotorula glutinis*. *Archives of microbiology*, 111(1-2):151–154, 1976.
- [66] Liling Jiang, Shengjun Luo, Xiaolei Fan, Zhiman Yang, and Rongbo Guo. Biomass and lipid production of marine microalgae using municipal wastewater and high concentration of CO₂. *Applied Energy*, 88(10):3336–3341, 2011.
- [67] Irving S Johnson. Human insulin from recombinant DNA technology. *Science*, 219(4585):632–637, 1983.
- [68] RH Jongbloed, JMAM Clement, and GWFH Borst-Pauwels. Kinetics of NH₄⁺ and K⁺ uptake by ectomycorrhizal fungi. effect of NH₄⁺ on K⁺ uptake. *Physiologia Plantarum*, 83(3):427–432, 1991.
- [69] Ugur Kaplan, Metin Türkay, L Biegler, and Bülent Karasözen. Modeling and simulation of metabolic networks for estimation of biomass accumulation parameters. *Discrete Applied Mathematics*, 157(10):2483–2493, 2009.
- [70] Elena Kazamia, David C Aldridge, and Alison G Smith. Synthetic ecology—a way forward for sustainable algal biofuel production? *Journal of Biotechnology*, 162(1):163–169, 2012.
- [71] Kamil A. Khan. *Sensitivity Analysis for Nonsmooth Dynamic Systems*. PhD thesis, Massachusetts Institute of Technology, 2015.
- [72] Kamil A. Khan and Paul I. Barton. Generalized derivatives for solutions of parametric ordinary differential equations with non-differentiable right-hand sides. *Journal of Optimization Theory and Applications*, 163(2):355–386, 2014.
- [73] Kamil A. Khan and Paul I. Barton. Switching behavior of solutions of ordinary differential equations with abs-factorable right-hand sides. *Systems & Control Letters*, 84:27–34, 2015.
- [74] Kamil A. Khan and Paul I. Barton. A vector forward mode of automatic differentiation for generalized derivative evaluation. *Optimization Methods and Software*, 30(6):1185–1212, 2015.
- [75] Kamil A. Khan and Paul I. Barton. Generalized derivatives for hybrid systems. *IEEE Transactions on Automatic Control*, 2017.
- [76] Diethelm Kleiner. The transport of NH₃ and HN₄⁺ across biological membranes. *Biochimica et Biophysica Acta (BBA)-Reviews on Bioenergetics*, 639(1):41–52, 1981.

- [77] Niels Klitgord and Daniel Segrè. Environments that induce synthetic microbial ecosystems. *PLoS Computational Biology*, 6(11):e1001002, 2010.
- [78] Masakazu Kojima and Susumu Shindo. Extension of Newton and quasi-Newton methods to systems of PC1 equations. *Journal of the Operations Research Society of Japan*, 29(4):352–374, 1986.
- [79] Qing-Xue Kong, Ling Li, Blanca Martinez, Paul Chen, and Roger Ruan. Culture of microalgae *Chlamydomonas reinhardtii* in wastewater for biomass feedstock production. *Applied Biochemistry and Biotechnology*, 160(1):9–18, 2010.
- [80] Alexei Kuntsevich and Franz Kappel. Solvopt: The solver for local nonlinear optimization problems. *Institute for Mathematics, Karl-Franzens University of Graz*, 1997.
- [81] Ruth E Ley, Peter J Turnbaugh, Samuel Klein, and Jeffrey I Gordon. Microbial ecology: human gut microbes associated with obesity. *Nature*, 444(7122):1022–1023, 2006.
- [82] Jiayin Ling, Saiwa Nip, Wai Leong Cheok, Renata Alves de Toledo, and Ho-jae Shim. Lipid production by a mixed culture of oleaginous yeast and microalga from distillery and domestic mixed wastewater. *Bioresource Technology*, 173:132–139, 2014.
- [83] David G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1989.
- [84] Rafael Luque. Algal biofuels: the eternal promise? *Energy & Environmental Science*, 3(3):254–257, 2010.
- [85] Bing Ma, Larry J Forney, and Jacques Ravel. Vaginal microbiome: rethinking health and disease. *Annual review of microbiology*, 66:371–389, 2012.
- [86] R. Mahadevan and C.H. Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic Engineering*, 5:264–276, 2003.
- [87] Radhakrishnan Mahadevan, Jeremy S. Edwards, and Francis J. Doyle III. Dynamic flux balance analysis of diauxic growth in *Escherichia coli*. *Biophysical Journal*, 83:1331–1340, September 2002.
- [88] L Mailleret, JL Gouzé, and O Bernard. Nonlinear control for algae growth models in the chemostat. *Bioprocess and biosystems engineering*, 27(5):319–327, 2005.
- [89] Marko M. Mäkelä. Survey of bundle methods for nonsmooth optimization. *Optimization Methods and Software*, 17(1):1–29, 2001.

- [90] Longfei Mao and Wynand S. Verwoerd. ORCA: a COBRA toolbox extension for model-driven discovery and analysis. *Bioinformatics*, 30(4):584–585, 2014.
- [91] Warren Lee McCabe, Julian Cleveland Smith, and Peter Harriott. *Unit operations of chemical engineering*. McGraw-Hill New York, 7th edition, 2005.
- [92] Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I: Convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.
- [93] Jonathan Monk, Juan Nogales, and Bernhard Ø Palsson. Optimizing genome-scale network reconstructions. *Nature biotechnology*, 32(5):447–452, 2014.
- [94] Jacques Monod. The growth of bacterial cultures. *Annual Reviews in Microbiology*, 3(1):371–394, 1949.
- [95] National Renewable Energy Laboratory. Photovoltaic solar resource of the United States. https://www.nrel.gov/gis/images/eere_pv/national_photovoltaic_2012-01.jpg, 2012.
- [96] Yu. Nesterov. Lexicographic differentiation of nonsmooth functions. *Mathematical Programming*, 104(2-3):669–700, 2005.
- [97] Jeffrey D Orth, Ronan MT Fleming, and Bernhard O Palsson. Reconstruction and use of microbial metabolic networks: the core *Escherichia coli* metabolic model as an educational guide. *EcoSal plus*, 2010.
- [98] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.
- [99] Basavaraj Palabhanvi, Vikram Kumar, Muthusivaramapandian Muthuraj, and Debasish Das. Preferential utilization of intracellular nutrients supports microalgal growth under nutrient starvation: Multi-nutrient mechanistic model and experimental validation. *Bioresource technology*, 173:245–255, 2014.
- [100] Bernhard Ø Palsson. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, New York, NY, 2006.
- [101] Seraphim Papanikolaou, Afroditi Chatzifragkou, Stylianos Fakas, Maria Galiotou-Panayotou, Michael Komaitis, Jean-Marc Nicaud, and George Aggelis. Biosynthesis of lipids and organic acids by *Yarrowia lipolytica* strains cultivated on glucose. *European Journal of Lipid Science and Technology*, 111(12):1221–1232, 2009.
- [102] Thidarat Papone, Supaporn Kookkhunthod, and Ratanaporn Leesing. Microbial oil production by monoculture and mixed cultures of microalgae and oleaginous yeasts using sugarcane juice as substrate. *World Acad Sci Eng Technol*, 64:1127–1131, 2012.

- [103] Liqun Qi and Jie Sun. A nonsmooth version of Newton’s method. *Mathematical programming*, 58(1):353–367, 1993.
- [104] Arvind U Raghunathan, J Ricardo Pérez-Correa, Eduardo Agosin, and Lorenz T Biegler. Parameter estimation in metabolic flux balance models for batch fermentation: Formulation & solution using differential variational inequalities (DVI). *Annals of Operations Research*, 148(1):251–270, 2006.
- [105] Colin Ratledge. Single cell oils: have they a biotechnological future? *Trends in Biotechnology*, 11(7):278–284, 1993.
- [106] I Rawat, R Ranjith Kumar, T Mutanda, and F Bux. Biodiesel from microalgae: a critical evaluation from laboratory to large scale production. *Applied Energy*, 103:444–467, 2013.
- [107] Jennifer L. Reed, Thuy D. Vo, Christophe H. Schilling, and Bernhard Ø. Palsson. An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). *Genome Biology*, 4(9):R54, 2003.
- [108] Raúl Reyna-Martínez, Ricardo Gomez-Flores, Ulrico J López-Chuken, Rosario González-González, Sergio Fernández-Delgadillo, and Isaias Balderas-Rentería. Lipid production by pure and mixed cultures of *Chlorella pyrenoidosa* and *Rhodotorula mucilaginosa* isolated in Nuevo León, México. *Applied Biochemistry and Biotechnology*, 175(1):354–359, 2015.
- [109] R.A. Robinson and R.H. Stokes. *Electrolyte Solutions*. Butterworths Scientific Publications, London, 1959.
- [110] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.
- [111] Liliana Rodolfi, Graziella Chini Zittelli, Niccolò Bassi, Giulia Padovani, Nataschia Biondi, Gimena Bonini, and Mario R Tredici. Microalgae for oil: Strain selection, induction of lipid synthesis and outdoor mass cultivation in a low-cost photobioreactor. *Biotechnology and Bioengineering*, 102(1):100–112, 2009.
- [112] Walter Rudin. *Principles of mathematical analysis*. 1976.
- [113] Mark F Ruth and Robert J Wooley. The cost of lignocellulosic sugar for commodity chemical production. In *4th Annual Green Chemistry & Engineering Conference*, page 109. Citeseer, 2000.
- [114] Nikolaos V Sahinidis. BARON: A general purpose global optimization software package. *Journal of global optimization*, 8(2):201–205, 1996.
- [115] Fabio Santamauro, Fraeya M Whiffin, Rod J Scott, Christopher J Chuck, et al. Low-cost lipid production by an oleaginous yeast cultured in non-sterile conditions using model waste resources. *Biotechnology for Biofuels*, 7:34–43, 2014.

- [116] Carla A Santos and Alberto Reis. Microalgal symbiosis in biotechnology. *Applied Microbiology and Biotechnology*, 98(13):5839–5846, 2014.
- [117] Jan Schellenberger, Junyoung O. Park, Tom M. Conrad, and Bernhard Ø. Palsson. BiGG: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC Bioinformatics*, 11(1):213, 2010.
- [118] Jan Schellenberger, Richard Que, Ronan M.T. Fleming, Ines Thiele, Jeffrey D. Orth, Adam M. Feist, Daniel C. Zielinski, Aarash Bordbar, Nathan E. Lewis, Sorena Rahmanian, Joseph Kang, Daniel R. Hyduke, and Bernhard Ø. Palsson. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. *Nature Protocols*, 6:1290–1307, 2011.
- [119] Stefan Scholtes. *Introduction to Piecewise Differentiable Equations*. Springer, New York, 2012.
- [120] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
- [121] Joseph K. Scott, Matthew D. Stuber, and Paul I. Barton. Generalized McCormick relaxations. *Journal of Global Optimization*, 51:569–606, 2011.
- [122] Michael L Shuler and Fikret Kargi. *Bioprocess engineering*. Prentice Hall New York, 2002.
- [123] SIA Partners. Energy outlook. Biofuels: how the United States became the world’s largest producer? <http://energy.sia-partners.com/biofuels-how-united-states-became-worlds-largest-producer>, February 2014. Accessed: 2016-12-09.
- [124] Peter Stechlinski, Kamil A Khan, and Paul I. Barton. Generalized sensitivity analysis of nonlinear programs. *SIAM Journal on Optimization*, Accepted.
- [125] Gregory Stephanopoulos, Aristos A Aristidou, and Jens Nielsen. *Metabolic engineering: principles and methodologies*. Academic press, 1998.
- [126] Systems Biology Research Group. Other organisms. <http://http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms>.
- [127] I. Tang, Martin R. Okos, Shang-Tian Yang, et al. Effects of pH and acetic acid on homoacetic fermentation of lactate by *Clostridium formicoaceticum*. *Biotechnology and Bioengineering*, 34(8):1063–1074, 1989.
- [128] The White House. Blueprint for a secure energy future. Report, 2011.
- [129] Gavin Towler and Ray K Sinnott. *Chemical engineering design: principles, practice and economics of plant and process design*. Elsevier, 2012.
- [130] Angelos Tsoukalas and Alexander Mitsos. Multivariate McCormick relaxations. *Journal of Global Optimization*, 59(2-3):633–662, 2014.

- [131] Mikio Tsuzuki. Mode of HCO_3^- -utilization by the cells of *Chlamydomonas reinhardtii* grown under ordinary air. *Z. Pflanzenphysiol. Bd.*, 110:29–37, 1983.
- [132] Richard Turton, Richard C Bailie, Wallace B Whiting, and Joseph A Shaeiwitz. *Analysis, synthesis and design of chemical processes*. Prentice Hall, 3rd edition, 2010.
- [133] United States Government Accountability Office. Alternative jet fuels: Federal activities support development and usage, but long-term commercial viability hinges on market factors. Report, 2014.
- [134] U.S. Energy Information Administration. Today in energy: Few transportation fuels surpass the energy densities of gasoline and diesel. <https://www.eia.gov/todayinenergy/detail.php?id=9991>, 2013.
- [135] U.S. Energy Information Administration. Electric power monthly. http://www.eia.gov/electricity/monthly/epm_table_grapher.cfm?t=epmt_5_6_a, 2015.
- [136] U.S. Environmental Protection Agency. Sources of greenhouse gas emissions. <https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions>, 2015.
- [137] Amit Varma and Bernhard Ø Palsson. Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* W3110. *Applied and environmental microbiology*, 60(10):3724–3731, 1994.
- [138] Heinrich Von Stackelberg. *The theory of the market economy*. Oxford University Press, 1952.
- [139] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.
- [140] Graeme M Walker. *Yeast physiology and biotechnology*. John Wiley & Sons, 1998.
- [141] Mark S Wigmosta, André M Coleman, Richard J Skaggs, Michael H Huesemann, and Leonard J Lane. National microalgae biofuel production potential and resource demand. *Water Resources Research*, 47(3):W00H04, 2011.
- [142] A Katharina Wilkins, Bruce Tidor, Jacob White, and Paul I Barton. Sensitivity analysis for oscillating dynamical systems. *SIAM Journal on Scientific Computing*, 31(4):2706–2732, 2009.
- [143] Peter J. le B. Williams and Lieve M.L. Laurens. Microalgae as biodiesel & biomass feedstocks: review & analysis of the biochemistry, energetics & economics. *Energy & Environmental Science*, 3(5):554–590, 2010.

- [144] Aidong Yang. Modeling and evaluation of CO₂ supply and utilization in algal ponds. *Industrial and Engineering Chemistry Research*, 50:11181–11192, 2011.
- [145] Hong-Wei Yen, Pin-Wen Chen, and Li-Juan Chen. The synergistic effects for the co-cultivation of oleaginous yeast-*Rhodotorula glutinis* and microalgae-*Scenedesmus obliquus* on the biomass and total lipids accumulation. *Bioresource Technology*, 184:148–152, 2015.
- [146] Guochang Zhang, William Todd French, Rafael Hernandez, Earl Alley, and Maria Paraschivescu. Effects of furfural and acetic acid on growth and lipid production from glucose and xylose by *Rhodotorula glutinis*. *Biomass and Bioenergy*, 35(1):734–740, 2011.
- [147] Xue-Wu Zhang, Feng Chen, and Michael R. Johns. Kinetic models for heterotrophic growth of *Chlamydomonas reinhardtii* in batch and fed-batch cultures. *Process Biochemistry*, 35:385–389, 1999.
- [148] Zhiping Zhang, Hairui Ji, Guiping Gong, Xu Zhang, and Tianwei Tan. Synergistic effects of oleaginous yeast *Rhodotorula glutinis* and microalga *Chlorella vulgaris* for enhancement of biomass and lipid yields. *Bioresource Technology*, 164:93–99, 2014.
- [149] K. Zhuang, E. Ma, Derek R. Lovley, and Radhakrishnan Mahadevan. The design of long-term effective uranium bioremediation strategy using a community metabolic model. *Biotechnology and Bioengineering*, 109:2475–2483, 2012.
- [150] Kai Zhuang, Mounir Izallalen, Paula Mouser, Hanno Richter, Carla Risso, Radhakrishnan Mahadevan, and Derek R. Lovley. Genome-scale dynamic modeling of the competition between *Rhodospirillum rubrum* and *Geobacter* in anoxic subsurface environments. *The ISME Journal*, 5:305–316, 2011.
- [151] Manfred Zinn, Bernard Witholt, and Thomas Egli. Dual nutrient limited growth: models, experimental observations, and applications. *Journal of biotechnology*, 113(1):263–279, 2004.