

Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

3-2003

Deterministic Intracellular Modeling

Jacqueline B. Young

Follow this and additional works at: <https://scholar.afit.edu/etd>



Part of the [Computer Sciences Commons](#), and the [Other Pharmacology, Toxicology and Environmental Health Commons](#)

Recommended Citation

Young, Jacqueline B., "Deterministic Intracellular Modeling" (2003). *Theses and Dissertations*. 4197.
<https://scholar.afit.edu/etd/4197>

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.



DETERMINISTIC INTRACELLULAR MODELING

THESIS

Jacqueline B. Young, Second Lieutenant, USAF

AFIT/GCS/ENC/03M-1

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCS/ENC/03M-1

DETERMINISTIC INTRACELLULAR MODELING

THESIS

Presented to the Faculty
Department of Mathematics and Statistics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Science

Jacqueline B. Young, BS
Second Lieutenant, USAF

March 2003

Approved for public release; distribution unlimited

DETERMINISTIC INTRACELLULAR MODELING

Jacqueline B. Young, BS

Second Lieutenant, USAF

Approved:

/signed/

Dennis W. Quinn (Chairman)

Date

/signed/

William P. Baker (Member)

Date

/signed/

Lawrence K. Chilton (Member)

Date

Acknowledgements

I praise God—The Creator. HE is my sole Sustainer.

I am highly appreciative for Dr. Quinn's continuous support in facilitating my understanding of my thesis topic towards successful completion and the support of Dr. Baker and Lt Col Chilton for their valuable suggestions which made this a better thesis.

Jacqueline B. Young

Table of Contents

	Page
Acknowledgements	iv
List of Figures	viii
List of Tables	ix
Abstract	xi
I. Introduction	1-1
1.1 Overview	1-1
1.2 Problem	1-1
1.3 Scope	1-2
1.4 Summary of Thesis	1-2
II. Background	2-1
2.1 Overview	2-1
2.2 Cellular Biology	2-4
2.2.1 Cell Types	2-4
2.2.2 Intracellular Structure	2-4
2.2.3 Eukaryotic Cell Division Cycle	2-5
2.2.4 Bacteriophage	2-7
2.2.5 <i>Vibrio fischeri</i>	2-7
2.3 Biochemistry	2-8
2.3.1 Proteins	2-8
2.3.2 Nucleic Acids	2-9
2.3.3 Chemical Reactions	2-9

	Page
2.3.4 Chemical Kinetics	2-10
2.4 Biological Pathways	2-17
2.5 Phosphorylation	2-18
2.6 Protein Synthesis	2-18
2.7 Mathematical Modeling	2-19
2.8 Modeling Approaches	2-21
2.9 Gepasi	2-24
2.10 Virtual Cell	2-25
2.11 E-CELL2	2-25
2.12 BioSPICE	2-26
III. Intracellular Modeling	3-1
3.1 Overview	3-1
3.2 Rate-Equation Approach	3-2
3.3 BioCharon	3-5
3.3.1 Bio Sketch Pad	3-5
3.3.2 Charon	3-7
3.4 JigCell	3-7
3.5 Biomolecular networks	3-8
3.6 Non-Biological Network: Modified Brusselator	3-8
3.7 Natural Networks	3-14
3.7.1 Lysis-Lysogeny Pathway	3-15
3.7.2 Vibrio fischeri	3-19
3.7.3 Budding Yeast Cell Cycle	3-24
3.7.4 Glycolysis	3-28

	Page
IV. Computational Model Results	4-1
4.1 Overview	4-1
4.2 Experimental Design	4-2
4.3 Modified Brusselator Model	4-4
4.3.1 General Results	4-4
4.3.2 Comparison Results	4-5
4.4 Lysis-Lysogeny Pathway	4-9
4.4.1 General Results	4-9
4.4.2 Simulation Results	4-13
4.5 <i>V. fischeri</i>	4-14
4.5.1 General Results	4-14
4.5.2 Comparison Results	4-21
4.6 Glycolytic Pathway	4-21
4.6.1 General Results	4-21
4.6.2 Comparison Results	4-26
4.7 Yeast Cell Cycle	4-27
4.7.1 General Results	4-27
4.7.2 Comparison Results	4-28
4.8 Summary	4-33
V. Summary and Conclusion	5-1
5.1 Summary	5-1
5.2 Conclusion	5-2
5.3 Recommendations	5-3
Appendix A. JigCell Model Builder Entries	A-1
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure		Page
2.1.	Eukaryotic Cell Division Cycle	2-6
2.2.	Protein Synthesis	2-20
3.1.	Simple Reaction Network	3-3
3.2.	BioCharon Software System	3-6
3.3.	Bio Sketch Pad	3-6
3.4.	Bioluminescence in <i>V. Fischeri</i>	3-23
3.5.	Budding Yeast Cell Division Cycle	3-27
3.6.	Glycolytic Pathway	3-30
4.1.	Autocatalytic Power Control (Modified Brusselator Model)	4-5
4.2.	Numerical Simulation I (Modified Brusselator Model)	4-6
4.3.	Numerical Simulation II (Modified Brusselator Model)	4-7
4.4.	Phase Plane Trajectories (Modified Brusselator Model)	4-8
4.5.	Conservation Relation Example (Lysis-Lysogeny Pathway)	4-12
4.6.	Population Size/Luminescence Correlation (<i>V. Fischeri</i>)	4-16
4.7.	Numerical Simulation I (<i>V. Fischeri</i>)	4-18
4.8.	Numerical Simulation II (<i>V. Fischeri</i>)	4-19
4.9.	Numerical Simulation III (<i>V. Fischeri</i>)	4-20
4.10.	Conservation Relation in Glycolytic Pathway)	4-25
4.11.	Numerical Simulation I (Budding Yeast Cell Division Cycle)	4-30
4.12.	Numerical Simulation II (Budding Yeast Cell Division Cycle)	4-31
4.13.	Numerical Simulation III (Budding Yeast Cell Division Cycle)	4-32
A.1.	<i>Model</i> Spreadsheet Entries (Lysis-Lysogeny Pathway)	A-2
A.2.	<i>Model</i> Spreadsheet Entries (<i>V. Fischeri</i>)	A-4
A.3.	<i>Model</i> Spreadsheet Entries (Budding Yeast)	A-7

List of Tables

Table		Page
3.1.	Species Cross Reference List (<i>V. fischeri</i>)	3-21
3.2.	Description of Model Parameters (<i>V. fischeri</i>)	3-22
3.3.	Cdk/CycB Molecular Interactions	3-25
3.4.	Species Cross Reference List (Glycolytic Pathway)	3-33
3.5.	Reaction Equations (Glycolytic Pathway)	3-34
3.6.	Reaction Equations with Catalytic Transitions (Glycolytic Pathway)	3-35
3.7.	Derived Reaction Velocities for Embden-Meyerhof Glycolytic Pathway	3-36
4.1.	Initial Conditions (Modified Brusselator Model)	4-9
4.2.	Critical Point Results (Modified Brusselator Model)	4-9
4.3.	Model Parameters (Lysis-Lysogeny Pathway)	4-10
4.4.	Critical Points (Lysis-Lysogeny Pathway)	4-10
4.5.	Eigenvalues for Full Lysis-Lysogeny Pathway	4-11
4.6.	Eigenvalues for Reduced Lysis-Lysogeny Pathway	4-11
4.7.	Initial Conditions (Lysis-Lysogeny Pathway)	4-14
4.8.	Comparison Results (Lysis-Lysogeny Pathway)	4-14
4.9.	Error (Lysis-Lysogeny Pathway)	4-15
4.10.	Initial Conditions (<i>V. fischeri</i>)	4-22
4.11.	Trial 1 Simulation Results (<i>V. fischeri</i>)	4-22
4.12.	Trial 2 Simulation Results (<i>V. fischeri</i>)	4-22
4.13.	Trial 3 Simulation Results (<i>V. fischeri</i>)	4-23
4.14.	Trial 4 Simulation Results (<i>V. fischeri</i>)	4-23
4.15.	Trial 5 Simulation Results (<i>V. fischeri</i>)	4-23

Table		Page
4.16.	Trial 6 Simulation Results (<i>V. fischeri</i>)	4-24
4.17.	Initial Conditions (Glycolytic Pathway)	4-27
4.18.	Conservation Relation Constants (Glycolytic Pathway)	4-28
4.19.	Comparison Results (Glycolytic Pathway)	4-29
4.20.	MATLAB $k_{d,20}$ Perturbation Results (Budding Yeast)	4-33
4.21.	JigCell $k_{d,20}$ Perturbation Results (Budding Yeast)	4-34
4.22.	Comparison Results (Budding Yeast)	4-34

Abstract

The United States Air Force is interested in the potential side effects—at the cellular level—from exposure to mission-essential chemicals. Presently, Air Force toxicology studies are conducted to help shed light in identifying potential hazards to workers. However, it takes a considerable amount of money, resources, and time to obtain and analyze experimental results from toxicology studies. The necessity for innovative methods that enable researchers to more effectively generate and analyze data is apparent.

Mathematical modeling is a viable option to become a valuable tool for the researcher. Mathematical models can rapidly generate informative predictions on how a cell reacts to a certain toxicant exposure. Moreover, information is readily available when generated by mathematical models.

This research involves the study of one non-biological reaction system and four biological, intracellular reaction systems. Each system is converted into a mathematical model using the rate-equation approach. Numerical simulation results from these mathematical models are obtained using two novel software modeling tools and MATLAB. Results obtained from the novel modeling tools are compared to MATLAB's results in order to ascertain the accuracy of each novel modeling tool.

The experience that is gained in deriving mathematical models and using novel tools to perform numerical simulations for these reaction systems should help the Air Force develop intracellular models to assist in future toxicology studies.

DETERMINISTIC INTRACELLULAR MODELING

I. Introduction

1.1 Overview

The potential of Air Force personnel being exposed to harmful chemicals within the work environment must be taken under serious consideration. This risk to personnel is heightened, because Air Force members are oftentimes required to work—over a wide spectrum of conditions—on or around specialized equipment that contain an assortment of toxic substances. Toxicity studies are conducted to help shed light in identifying potential hazards to workers. However, the cost of toxicity studies is high.

Performing toxicity studies are formidable for two reasons. Experiments are very costly in funds and resources. Also, it usually takes a considerable amount of time to obtain and analyze experimental results. The necessity for innovative methods that enable researchers to more effectively generate and analyze data is apparent.

Mathematical modeling has the potential to become a valuable tool for the researcher. Credible mathematical models can generate informative predictions on how a cell reacts to being exposed to a certain toxicant. Moreover, information is readily available when produced by mathematical models.

1.2 Problem

In order for the Air Force to develop mathematical models that can be ultimately coupled with existing experimental data on the cell, it is important to understand how various intracellular processes are regulated. By deriving computational

models from well-documented cellular systems and implementing those models on novel software tools, valuable experience is gained. This newfound knowledge can be used to model cellular systems of a higher order. This knowledge will lead to a better understanding of how the Air Force should construct intracellular models to assist future toxicity studies.

1.3 Scope

This research involves a literature review on biological topics relevant to the living cell. This review includes microbiology, biochemistry, chemical kinetics, and intracellular structure. Mathematical models (deterministic) are developed from synthetic and naturally occurring networks. These models are subsequently simulated on three software applications: MATLAB, BioCharon ([2] and [5]), and JigCell ([31] and [30]). Results from these software applications are analyzed.

1.4 Summary of Thesis

This thesis is organized as follows:

Chapter 2 gives an overview on cellular biology and biochemistry. Then, mathematical modeling is formally introduced. Three alternative approaches for mathematical modeling are briefly discussed. It concludes with a discussion on existing software tools that can be used for intracellular modeling.

Chapter 3 gives a detailed description of the rate-equation approach (a mathematical modeling method). Three software applications—MATLAB, JigCell, and BioCharon—that can be used to model intracellular processes are briefly discussed. One non-biological network and four naturally occurring networks are presented. Computational models for the synthetic and three naturally occurring networks are explicitly derived.

Chapter 4 presents general simulation results and comparisons for the reaction systems discussed in chapter 3. Both general and comparison results are analyzed.

Chapter 5 summarizes the work completed. It gives conclusions and recommendations for future work.

II. Background

2.1 Overview

“Cells are the fundamental working units of every living system [29:1].” They encase a dynamic environment of ongoing activities, such as synthesis of proteins or the breakdown of glucose. The underlying logic for cellular function is highly complex. Understanding this functionality is the bedrock for conceptualizing and comprehending higher order phenomena, such as physiology, anatomy, and ecology [14:544].

The dynamic environment within the cell (intracellular) involves a sophisticated degree of interaction between three important classes of macromolecules: deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and proteins [14:544]. These molecules continuously interact by means of biochemical reactions. Some of these reactions take place independently and others take place serially. As a result, molecules within the cell are in constant flux. This flux is oftentimes regulated by genes.

Genes—specified segments of DNA—control intracellular processes by specifying the synthesis of enzymes (a special form of a protein) and other types of proteins [9:6]. Specific groups of genes may be activated or ‘expressed’ by particular signals in order to regulate a common cell process. Moreover, groups of genes may modulate expression levels of other genes. Such groups are called *genetic regulatory networks* [26:248].

These types of networks are commonly divided into separate modules of operations. In this way, cell functionality can be viewed “as a collection of interrelated subsystems [14:544].” Nonetheless, describing these networks—especially quantitatively—is a monumental undertaking. Smolen et al. elaborate on the inherent complexity of genetic regulatory networks: “Understanding the combined effects of these phenomena is often beyond the capacity of intuition [26:249].”

A common approach to begin the elucidation of cellular functionality is through modeling methods. Models describing cellular networks must embody hypotheses made about each subsystem as well as on how these subsystems are subsequently integrated back together. Presently, mathematical modeling—via mapping a cellular network to a system of ordinary differential equations (ODEs)—is widely used to describe the dynamics of a cellular network. Hasty et al. describe the process of implementing mathematical models [10:277]:

The modeling of gene regulatory networks relies on characterization of the behavior of small subsystems, formation of hypotheses about how these subsystems interconnect, translation of these hypotheses into a *mathematical model* and experimentation to yield results that indicate necessary changes to the original hypotheses.

Mathematical modeling has two major advantages. “The *precision* of the mathematical language makes mathematical modeling a useful framework for conceptualizing and understanding complex biochemical systems [26:248].” Consequently, the modeler is able to precisely define hypotheses made about each network subsystem and overall network organization by a well-defined language. Most importantly, present circumstances (explained below) are primed for quantitative as well as qualitative coupling to experimental data.

Since large amounts of data at the genomic level are readily available and experimental methods are continually being enhanced, the stage is set for building models that make physical sense. Consequently, the modeler is able to ‘verify’ candidate models. Plus, the modeler can make ‘informed’ modifications to model parameters (e.g., change values of kinetic rate constants), based on experimental data. In this way, mathematical models have the potential to be tightly coupled to experimental data.

For example, the modeler can compare a temporal data set for a natural biological system to model predictions. Discrepancies between the data set and model predictions are a strong indication that the proposed model contains errors. The

modeler can then refine or even reformulate certain hypotheses made about network architecture and exactly implement those changes in the mathematical model. Since formulating a computational model involves a series of intricate steps—often overwhelming the pencil-and-paper method—there is a necessity to develop software packages that assist modelers in the modeling process.

Fortunately, there are already existing software packages for modeling biochemical networks. In general, these software packages add a beneficial layer of abstraction between a schematic or *wiring diagram* that describes the architecture or pathways of a biochemical network and the corresponding computational model generated from that diagram. This layer of abstraction is often incorporated by means of a graphical user interface (GUI). The user enters details about the biochemical network in symbolic form (e.g., graphical objects or chemical reaction notation). Then, the software generates the computational model from this input. In this fashion, the user is spared the intricacies of explicitly transforming (mapping) the schematic or wiring diagram to its corresponding computational model. Consequently, the likelihood of making errors in deriving computational models is usually reduced, especially when attempting to model large networks.

As a precursor for using available software packages, the modeler must have an understanding of the fundamentals of life sciences related to cellular networks, such as microbiology, biochemistry, and genetics. Since mathematical modeling generally requires derivation of a system of nonlinear ordinary differential equations (ODEs) from a wiring diagram, the modeler should also be well-grounded in mathematical analysis of nonlinear ODEs. From a mathematical (technical) perspective, the modeler is now able to predict and explain subsequent model behavior. In the following sections, I briefly elaborate on these essential topics that serve as a prerequisite for modeling cellular networks.

2.2 Cellular Biology

Understanding cellular structure is essential in implementing the correct approach to model cellular behavior. Cellular functionality is generally partitioned into various components of the cell. In order to model mechanisms of cellular behavior, these cellular components must be understood. This section contains references to intracellular structure for prokaryotic and eukaryotic cell types.

2.2.1 Cell Types. All living cell types can be classified as either prokaryotic or eukaryotic [6:71]. The structural complexity of prokaryotes is simpler than eukaryotes. Organisms in the prokaryotic class consist of only one cell, such as bacteria. They “lack a nucleus and other membrane-enclosed structures [6:71].” On the other hand, eukaryotes encompass all plants, animal, fungi and protists [6:71]. Structures in this class are more defined. For example, cells in this class possess a nucleus and other membrane-enclosed structures. Related material about prokaryotic and eukaryotic cell types can be found in [6:71–95].

2.2.2 Intracellular Structure. Using bacteria cells as the prokaryotic prototype, these cells include a cell membrane, cytoplasm, ribosomes, and a nuclear region. Cytoplasm of prokaryotic cells is the semifluid substance inside the cell membrane. Ribosomes consist of ribonucleic acid and protein. Finally, the nuclear region has DNA arranged in one large, circular chromosome [6:103].

The structure for eukaryotic cells is more complex than prokaryotic cells. These type of cells include a membrane-enclosed cell nucleus, with a nuclear envelope, nucleoplasm, nucleoli, and chromosomes [6:103]. The following list outlines structural components of prokaryotic cells [6:91–92].

- Cytoplasm contains elements of a cytoskeleton, a fibrous network that give support and shape these cells.

- Cell nucleus is a distinct organelle with a nuclear envelope, nucleoplasm, nucleoli, and chromosomes.
- Nuclear envelope consists of a double membrane, each layer of which is structurally like the plasma membrane.
- Nucleoplasm is the semifluid portion of the nucleus.
- Nucleoli contain a significant amount of RNA and serve as sites for the assembly of ribosomes.
- Chromosomes—typically paired—contain DNA and proteins called histones.
- Histones contribute directly to the structure of chromosomes.

More information about intracellular structure can be found in [6:103 - 104].

2.2.3 Eukaryotic Cell Division Cycle. “To reproduce itself, a cell must duplicate all its components and separate them, more or less evenly, to two daughter cells, so that each daughter has the information and machinery necessary to repeat the process. In general, eukaryotic cells replicate and partition their genetic material in two distinct, coordinated processes [8:369].” These two phases are described below.

S phase The DNA molecule in each chromosome is precisely replicated to form two identical sister chromatids that are held together by cohesins (tethering proteins).

M phase The cell builds a mitotic spindle, condenses its replicated chromosomes, aligns them on the midplane of the spindle, and then, at anaphase, removes the cohesins and separates sister chromatids to opposite poles of the spindle [8:369].

These two phases are separated temporally by gaps (G1 and G2 phases) [8:369]. Figure 2.1 depicts the eukaryotic cell cycle in general.

Several mechanisms are in place to coordinate the proper execution of events during the cell division cycle. For example, licensing factors (Mcm2–7 and Cdc6)

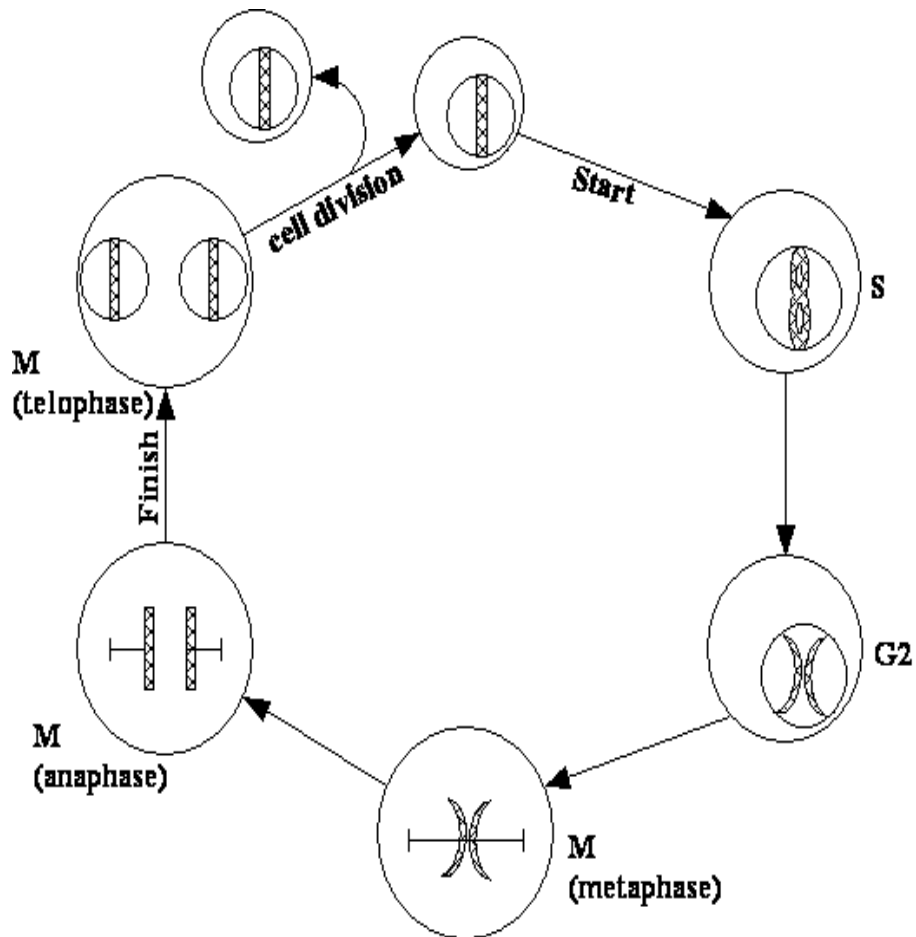


Figure 2.1. This diagram of the typical eukaryotic division cell cycle is based on a figure presented in [20:2]. Metaphase is when chromosomes are aligned between two mitotic spindles. Anaphase is when the glue that holds the sister chromatids together is dissolved, allowing each chromatid to be pulled by the microtubules to one of the poles of the spindle. Telephase is when two nuclei are created just before the cell divides (called cytokinesis).

make sure that each DNA molecule is replicated only once per cycle. Also, cycle logic does not allow the cell to commence anaphase until DNA replication is complete and each pair of sister chromatids is properly aligned [8:369]. Specific information on cell cycle control mechanisms can be found in [8:369–374].

2.2.4 Bacteriophage. Bacteriophages or phages “infect bacterial cells (hosts) and reproduce within them [6:200]”. Once the nucleic acid of the phage enters the bacterial cell, “further events follow one of two pathways, depending on whether the phage is virulent or temperate [6:200].” An infectious virulent phage is capable of causing the death of a bacterial cell. “When the cell becomes filled with a hundred or more phages, phage enzymes rupture the cell, releasing newly formed phages, which can then infect other cells [6:200].” This rupture of the infected cell is called a lytic cycle. On the other hand, “a temperate phage ordinarily does not cause a disruptive infection. Instead the phage DNA is incorporated into a bacterium’s DNA and is replicated with it [6:200].” More information on phages can be found in [6:200–203].

2.2.5 Vibrio fischeri. “*Vibrio fischeri* is a marine bacterium found both as a free-living organism and as a symbiont of some marine fish and squid [1:8].” *V. fischeri*, a single-celled organism, has a luminescence ability that is directly effected by the number of other *V. fischeri* cells residing at a close proximity (i.e., local population). As a free-living organism, population size is generally small, and luminescence appears to be absent. However, as a symbiont, population size is generally large (dense) and, luminescence is usually detectable.

Luminescence in *V. fischeri* is activated by its quorum sensing system. When the local population reaches a quorum—a minimum population size—the quorum sensing system activates a set of genes that enable the cell to become luminescent [1:8]. For information concerning genes, see section 2.3.2. More information about *V. fischeri* can be found in [1:7–8].

2.3 Biochemistry

Biochemistry is the study of chemical reactions that occur in living systems. Biochemistry focuses specifically on the molecules of matter (biomolecules), which living organisms are composed of. Four main classes of important biomolecules are proteins, carbohydrates, lipids, and nucleic acids [27:1]. Key objects in the modeling process are encompassed in this field of study.

Dynamic behavior of the intracellular environment is characterized by flux of various types of molecules (species) that are changed by biochemical reactions. Understanding the principles of biochemistry is essential in being able to incorporate this dynamic behavior into candidate models. Some key principles of biochemistry are the functionality of proteins and nucleic acids, how biochemical reactions take place, and how chemical kinetics govern reactions. I discuss these topics in the following sections.

2.3.1 Proteins. In the book, Evolutionary Computation in Bioinformatics, the author describes the significance of protein molecules, “Proteins make up most of an organism’s biomass, and play a key role in its metabolic and other cellular and bodily processes [9:12].” Specifically, a protein is a linear chain of amino acids. The amino-acid sequence of the protein, along with its three-dimensional shape, is the functional portion of information flow in a cell [14:545]. An enzyme is a special type of protein that is crucial in enabling chemical reactions to take place.

An enzyme is a catalyst for chemical reactions in living organisms [27:9]. During chemical reactions, a specific enzyme associates with a corresponding substrate—forms a complex—changes it into a product, and then dissociates with the new product. However, the enzyme remains unchanged and is ‘free’ to take part in another chemical reaction with a specific substrate. Enzyme specificity is related to its shape. Detailed information on proteins and enzymes can be found in [6:41–42] and [27:9–18].

2.3.2 Nucleic Acids. Nucleic acids are the carriers of the genetic code. “They contain genetic information that determines all the heritable characteristics of a living organism, be it a microbe or a human. Such information is passed from generation to generation and directs the protein synthesis in each organism [6:43].” RNA and DNA are the two nucleic acids found in living organisms [6:44]. The data structure for genetic information storage within each cell is stored in the following hierarchical configuration—genome, chromosomes, and genes.

The genome contains the complete set of instructions for an organism. “It contains the master blueprint for all cellular structures and activities for the lifetime of the cell or organism [13:5].” The human genome is subdivided into structures called chromosomes.

Chromosomes consist of tightly coiled threads of DNA and associated protein molecules. Apart from reproductive cells and mature red blood cells, every cell in the human body contains 23 pairs of chromosomes. Each chromosome is a packet of compressed and entwined DNA. The human genome contains three billion base pairs [13:5]. Subunits of a chromosomes are genes.

Genes are the basic physical and functional units of heredity. They are characterized as a “specific sequence of nucleotide bases, whose sequences carry the information required for constructing proteins, which provide the structural components of cells and tissues as well as enzymes for essential biochemical reactions [13:7].” The human genome contains about 30,000 to 40,000 genes [1:2]. Specific information on nucleic acids can be found in [13:5–9] and [6:42–45].

2.3.3 Chemical Reactions. Chemical reactions are continuously taking place within a cell. As a result, a cell is like a factory, where products are continuously being produced. Of course, this production is highly coordinated. As indicated in the following quote, enzyme regulation is a main factor in regulating chemical reactions [6:109].

Like nearly all other chemical processes in living organisms...each consist of a series of chemical reactions in which the product of one reaction serves as the substrate (reacting material) for the next. Each reaction in a pathway is controlled by a particular enzyme.

Hasty et al. discusses the regulation of intracellular chemical reactions from a standpoint of gene regulatory networks. They explicitly define negative and positive feedback as regulatory modes within a cellular network. Negative feedback is when a cellular network inhibits in own level of activity. Contrarily, positive feedback is when such a network increases its own level of activity [10:269–270]. The models of *Vibrio fischeri* (a unicellular bacteria) and *saccharoyces cerevisiae* (yeast cell) are covered in this thesis. They explicitly use positive and negative feedback to regulate the production of end products.

2.3.4 Chemical Kinetics. Intracellular chemical reactions occur at various rates involving a wide diversity of molecular interactions. Consequently, RNA and proteins are in flux. In order to build meaningful intracellular network models, this flux must be captured in candidate models. Thus, a fundamental understanding of chemical kinetics is necessary. In this section, some chemical kinetic terms—chemical reaction equation notation, reaction rates, and reaction rate laws—are briefly discussed.

A chemical reaction can be assigned a specific *reaction rate* (velocity) at which that chemical reaction takes place. Reaction rate is defined “as the change in concentration per unit of time of any reaction product for which the stoichiometric coefficient is 1 (assuming the volume of the reacting system remains constant) [25:4].” Stoichiometry is “the determination of the proportions in which chemical elements combine or are produced and the weight relations in any chemical reaction [19:1320].”

Various units of measure are used to quantify *concentration* in chemical reactions. Some common units of measure include moles per liter ($\frac{\text{mole}}{\text{l}}$), moles per

cubic centimeter ($\frac{mole}{cm^3}$), and molecules per cubic centimeter ($\frac{molecules}{cm^3}$) [25:4]. Plus, the second is generally the time unit.

For example, I generalize an example taken from [25:5]. The following one-way reaction in some unit of measure is assumed at a constant volume and obtain a rate of change of B_2 concentration of $\frac{1.5mole}{10^5lsec}$.



Then that is the rate of the reaction. From stoichiometry, the other reaction rates can be deduced.

$$\frac{d[A_2]}{dt} = \frac{1.5mole}{10^5lsec} \quad (2.2)$$

and

$$\frac{d[AB_2]}{dt} = \frac{-3.0mole}{10^5lsec} \quad (2.3)$$

In general, “for any reaction product, the rate of change of concentration is equal to the stoichiometric coefficient times the rate of reaction; while for the reactants, the rate of change of concentration is equal to the negative of the stoichiometric coefficient times the rate of reaction [25:5].”

Furthermore, the first equation can be rearranged so that the substances are all on the left hand side, with negative stoichiometric coefficients for the reactions.

$$A_2 + B_2 - 2AB = 0. \quad (2.4)$$

In the generalized form, this equation can be written as

$$\sum \nu_i A_i = 0, \quad (2.5)$$

where A_i is the symbol of a product or reactant, and ν_i is its stoichiometric coefficient. Plus, ν_i is positive for products and negative for reactants. Now, with this

redefinition of the stoichiometric coefficients and R representing the rate of reaction, then

$$\frac{d[A_i]}{dt} = \nu_i R. \quad (2.6)$$

Equations (1.2) and (1.3) can be derived using equation (1.4). In this case, both stoichiometric coefficients equal one and $R = \frac{1.5 \text{ mole}}{10^5 \text{ lsec}}$.

When building intracellular network models, a specific reaction rate law is often assumed for each reaction. At a given temperature, and perhaps within a limited range of concentrations, one can write a rate law for a reaction of the form,

$$k \prod_i [A_i]^{\alpha_i} [X_j]^{\beta_j} \quad (2.7)$$

where the A_i are the reactants, the X_j are substances that are not reactants but do influence the rate, the α 's and β 's are coefficients, that are not necessarily related to the stoichiometric coefficients ν , and k is a rate constant [25:9]. However, in this thesis I will only consider elementary reactions. Therefore, "for many elementary reactions, the rate law coefficients are equal to the stoichiometric coefficients [25:9]." Specifically, only mass action and Michaelis-Menten rate laws are implemented in this thesis.

Mass action rate laws are expressed in the form of equation 2.7. However, for mass action rate laws, α 's and β 's are stoichiometric coefficients for their respective substances. Unlike this rate law, the Michaelis-Menten kinetic rate law explicitly accounts for reactions catalyzed by enzymes.

"Most biological reactions are modeled by the Michaelis-Menten kinetic scheme [9:258]." Michaelis-Menten rate law is characterized by

$$V = \frac{k_2[S][E]_0}{[S] + K_m}, \quad (2.8)$$

where S is the reactant or substrate molecule, $[E]_0$ is the amount of enzymatic concentration, K_m is the Michaelis constant, and k_2 is the rate constant for the combination of an enzyme and an (unreacted) substrate molecule (ES) becoming dissociated [25:131].

Closely following [25:130–131], the Michaelis-Menten equation is derived with the assumption that the concentration of $[E]$ is significantly smaller than $[S]$ (i.e., $[E] \ll [S]$). Since $[E] \ll [S]$, a steady state concentration of $[ES]$ will quickly build up. Therefore, rate for this reaction is $V = k_2[ES]$. This equation depicts the appearance of the product P as E catalyzes S (discussed below).

The object in deriving the Michaelis-Menten equation is to represent V in terms of experimentally measurable quantities of $[S]$ and $[E]_0$. This is accomplished in the following derivation.

The author starts with a simple unimolecular reaction,



where S is the substrate molecule, and P is the product of the reaction. This reaction can then be divided into two different stages, represented by the following pair of reaction equations.



where k_1 is the forward rate constant and k_{-1} is the reverse reaction rate constant.



where k_2 is the forward reaction rate constant.

In equation 2.10 (moving left to right), S reversibly associates with E . In equation 2.11—the enzyme-substrate complex ES —breaks down into the original

enzyme and a product. E can be used over and over again to catalyze additional S molecules. This equation represents appearance of P after E catalyzes S .

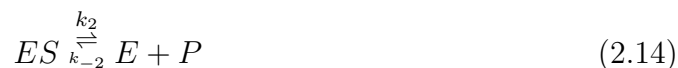
If equation 2.9 is a two-way reaction instead of a one-way reaction, the following can be deduced from equations 2.9, 2.10, and 2.11.



Again, this reaction is divided into two different stages, represented by the following pair of reaction equations.



where k_1 is the forward reaction rate constant and k_{-1} is the reverse reaction rate constant.



where k_2 is the forward reaction rate constant and k_{-2} is the reverse reaction rate constant.

This new set of equations is the same as the above set with the exception of 2.14. In this case, equation 2.14 is reversible. As a result, enzyme and a product— E and P —can bind back together to again form the complex ES .

Using equations 2.10 and 2.11 and applying standard techniques of biochemical kinetics ([25:8–10]), the following reaction velocities can be expressed,

$$v_1 = -k_1[E][S] + k_{-1}[ES] \quad (2.15)$$

$$v_2 = -k_2[ES] \quad (2.16)$$

where each term in equation 2.15 is obtained using equation 2.7. Therefore, the rate of change for $[ES]$ can be expressed as follows.

$$\begin{aligned}\frac{d[ES]}{dt} &= -v_1 + v_2 \\ &= k_1[E][S] - k_{-1}[ES] - k_2[ES]\end{aligned}\quad (2.17)$$

There is also a conservation relationship between the total concentration of enzyme $[E]_0$ and the sum of the concentration of enzyme associated with the substrate $[ES]$ and free enzyme $[E]$. This relationship is expressed as

$$[E]_0 = [ES] + [E] \Rightarrow [E] = [E]_0 - [ES] \quad (2.18)$$

Since $[E] \ll [S]$, this implies a steady state concentration of $[ES]$ will quickly occur [25:130]. This means that the rate of the overall reaction is

$$V = k_2[ES]. \quad (2.19)$$

Therefore, $[ES]$ remains constant, which implies $\frac{d[ES]}{dt} = 0$. Therefore,

$$0 = k_1[E][S] - k_{-1}[ES] - k_2[ES] \quad (2.20)$$

$$[ES](k_{-1} + k_2) = k_1[E][S]$$

$$[ES] = \frac{k_1[S]}{(k_{-1} + k_2)}[E]$$

$$[ES] = \frac{k_1[S]}{(k_{-1} + k_2)}([E]_0 - [ES])$$

$$[ES](k_{-1} + k_2) = k_1[S]([E]_0 - [ES])$$

$$[ES](k_{-1} + k_2 + k_1[S]) = k_1[S][E]_0$$

$$[ES] = \frac{k_1[S][E]_0}{(k_{-1} + k_2 + k_1[S])}$$

Substituting this into equation 2.19, in which the velocity of the build up of $[ES]$ is expressed, reduces to the following equation.

$$V = k_2 \frac{k_1[S][E]_0}{(k_{-1} + k_2 + k_1[S])} = \frac{k_2[S][E]_0}{[S] + (k_{-1} + k_2)/k_1} \quad (2.21)$$

The author then expresses the second term in the denominator as $K_m = (k_{-1} + k_2)/k_1$ —the Michaelis constant. Substituting K_m into equation 2.21 completes the derivation of the Michaelis-Menten equation. Additionally, the author discusses limiting behavior at low and high substrate concentrations exhibited by the Michaelis-Menten equation [25:131]. At low values of $[S]$,

$$V = \frac{k_2[S][E]_0}{K_m} \quad (2.22)$$

which gives a first-order dependence on $[S]$. In this case, there are a lot of ‘free’ enzyme molecules. As $[S]$ is increased, the rate of reaction is increased, up until all enzyme molecules are associated with substrate molecules [25:131].

On the other hand, at high levels of $[S]$, this term rather than K_m dominates the denominator to give

$$V = k_2[E]_0 \quad (2.23)$$

In this case, the rate of reaction no longer depends on $[S]$. When this state occurs, all enzymes are associated with substrate molecules (no free enzymes). Hence, the reaction has reached its maximum velocity (V_{max}). Additional substrate molecules will not be catalyzed until enzymatic molecules disassociate with substrate molecules currently undergoing chemical change [25:131].

This Michaelis-Menten equation is implemented in chapters three and four. Using the rate equation approach (discussed later), it is expressed in several reaction equations of the budding yeast cell cycle model.

2.4 *Biological Pathways*

A metabolic pathway consists of a series of chemical reactions in which the product of one reaction serves as the substrate (reacting material) for the next chemical reaction. Each reaction in a metabolic pathway is controlled by a particular enzyme. For example, $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ represents a simple metabolic pathway. In this pathway, A is the initial substrate, E is the final product, and B , C , and D are intermediates [6:109].

Metabolic pathways are either catabolic or anabolic (biosynthetic). Catabolic pathways capture energy in a form cells can use. Anabolic pathways make the complex molecules that form the structure of cells, enzymes, and other molecules that control cells. These pathways use building blocks such as sugars, glycerol, fatty acids, amino acids, nucleotides, and other molecules to make carbohydrates, lipids, proteins, nucleic acids, or a combination of these molecules. Adenosine triphosphate (ATP) molecules are the links that couple catabolic and anabolic pathways. Energy released in catabolic reactions is captured and stored in the form of ATP molecules,

which are later broken down to provide the energy needed to build up new molecules in biosynthetic pathways [6:109]. Glycolysis is an example of a metabolic pathway [6:114]. The glycolytic pathway model in section A incorporates ATP molecules.

2.5 Phosphorylation

“Phosphorylation is the addition of a phosphate group to a molecule, often from ATP. This addition generally increases the molecule’s energy. Thus, phosphate groups commonly serve as energy carriers in biochemical reactions [6:116].” In the glycolytic pathway model in section A, phosphate groups from two molecules of ATP are added to glucose at the beginning part of the pathway. This expenditure of two ATPs raises the energy level of glucose, enabling it to then participate in subsequent reactions [6:116].

Adenosine diphosphate (ADP) has a direct correlation with ATP in regards to glycolysis. With ADP and inorganic phosphate (P_i) available in the cytoplasm, the energy released from substrate molecules is used to form high-energy bonds between ADP and P_i :



In this way energy is captured in ATP at the substrate level [6:116]. Phosphorylation is a key activity in the glycolytic pathway and in the budding yeast cell cycle (*Saccharomyces cerevisiae*).

2.6 Protein Synthesis

With an understanding of the fundamental principles of Biochemistry, it is feasible to analyze one of the most important activities of intracellular function—protein synthesis. As previously mentioned, proteins are crucial components in major cell functions. The degree of importance is quantified by [6:169]: “All cells must constantly synthesize proteins to carry out their life processes: reproduction, growth, repair, and regulation.” Protein synthesis involves several orchestrated steps: the

gene (or groups of genes) is tagged for transcription (initialized), the gene is transcribed to a messenger RNA (mRNA), and mRNA interacts with a ribosome in building a specified protein from a sequence of amino acids previously prescribed by the gene [9:11]. In this manner, a gene or groups of genes indirectly build specified proteins.

Protein synthesis is initiated when “a pre-initiation complex is assembled around the promoter region just upstream of the gene that is to be expressed [9:11].” This complex attracts RNA polymerase, a protein that causes the strands of DNA to separate near the start of the gene. Moving along this gene or exon (the active part of DNA transcribing mRNA) RNA polymerase affects production of an RNA transcript [9:11]. Next, this mRNA transcript interacts with a ribosome—the location where protein is synthesized via translation of mRNA. In prokaryotic cells, transcription and translation takes place both in the cytoplasm, whereas in eukaryotes, transcription takes place within the cell nucleus, while translation takes place outside the nucleus [6:171]. Based on [6:176], the following steps summarize translation.

1. An mRNA transcript becomes properly oriented on a ribosome.
2. A ribosome reads each codon of the mRNA, and the appropriate tRNA combines with it and delivers a particular amino acid to the ribosome (a codon is a sequence of three bases in mRNA that specifies a particular amino acid [6:G13]).
3. Step 2 is repeated until a stop codon is read that terminates protein synthesis.

Using a diagram in [9:13] as a reference, Figure 2.2 illustrates how a protein is generated from genetic information.

2.7 Mathematical Modeling

“Mathematical models are useful for providing a framework for integrating data and gaining insights into the static and dynamic behavior of complex biological systems [26:247]....” This class of models approximates the dynamics of biological

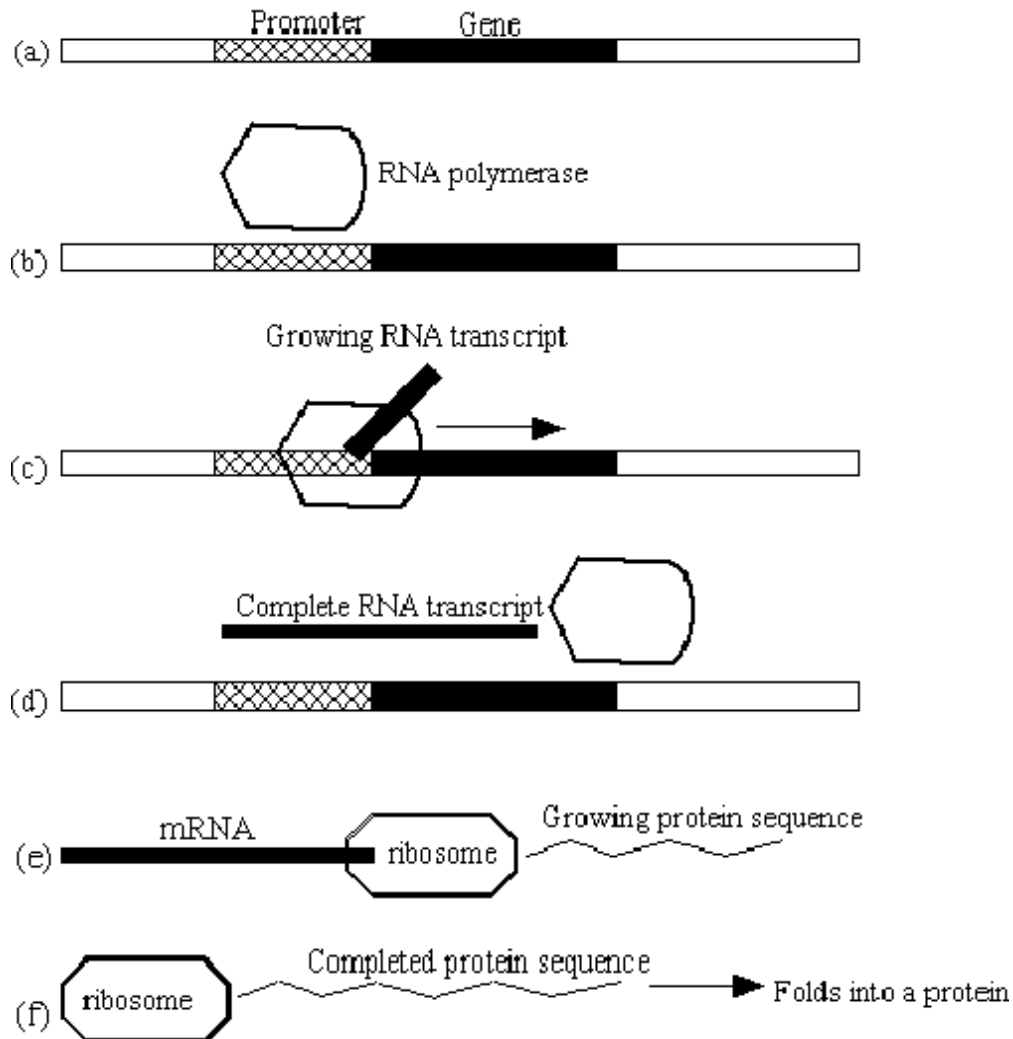


Figure 2.2. Steps (a)–(f) illustrate how a protein is made from genetic information (transcription/translation). (a) Section of DNA containing a gene, preceded by a promoter region. (b) An RNA polymerase complex is attracted to the DNA around the promoter and begins to interact with the DNA at the start of the gene—beginning of transcription. (c) The RNA polymerase moves along from left to right, gradually building an RNA transcript for the gene sequence. (d) The transcript is complete, and it now separates from both the RNA polymerase and the DNA. (e) The ribosome uses it to automatically manufacture a protein molecule—translation in progress. (f) Translation is complete. This diagram is based on an illustration published in [9:13].

systems using a system of ODEs (typically nonlinear) as the computational model. In addition, the system of ODEs can be augmented with logic terms that capture events (e.g., cell division). Hasty et al. defines the following states and regulative mechanisms of biological networks that have been thoroughly studied [10:269–270]. It is necessary to have an understanding of these terms, because these traits and regulative mechanisms are normally incorporated into candidate models.

Equilibrium state: For a gene product (such as a protein), the rate of production is balanced by its rate of degradation [10:269].

Fixed point: A point at which the rates of change of all variables in a system are exactly zero. A system precisely at its fixed point (or steady state) will remain there permanently [10:269].

Multistability: The property of having more than one stable fixed point [10:269].

Negative feedback: A component (species) of a system is subject to negative feedback when it inhibits its own level of activity [10:269].

Positive feedback: A component (species) of a system is subject to positive feedback when it increases its own level of activity [10:270].

Certain configurations of negative and positive feedback can lead to system multistability. This is an important concept for the modeler to grasp, because “understanding how multistability arises is thus relevant to understanding the operation of natural biological switches, as well as the design of synthetic switching networks [10:269].” Specific details on multistability and its implications can be found in [10:269].

2.8 *Modeling Approaches*

Once the modeler has obtained a well-rounded understanding on the basics of cellular structure, biochemistry, and mathematical modeling concepts, the modeler is nearly ready to consider a modeling approach for implementation. Prior to selecting

a modeling approach, the modeler should be familiar with modeling alternatives. I discuss three alternative modeling approaches: Boolean, chemical kinetics or rate equation, and stochastic [10:270]. These approaches range from low to high level of detail in characterizing a biological network.

Starting from a low level of network detail, the Boolean approach enables control of network dynamics by assigning a switch to each gene. In this way, each gene can assume only one of two states—ON or OFF. Also, genes within the network have the ability to interact with each other through user-defined interaction rules [10:270].

For example, suppose three genes—A, B, and C—collectively regulate a network. A possible rule for network dynamics could be: if genes A and B are OFF at the preceding time step, then gene C is turned ON during the current time step. This represents synchronous control of genetic states. Asynchronous methods for activating and deactivating gene switches can also be incorporated.

The advantage of the Boolean approach is that such models are not difficult to implement. Also, the computational cost to run simulations is relatively low. The disadvantage of this approach is that “the abstraction of genes to ON/OFF switches makes it difficult or impossible to include many of the details in cellular biology [10:270]”.

The chemical kinetics or rate equation approach approximates a cellular network at a midrange level of detail. As stated in section 2.7, a system of ODEs (typically nonlinear) approximates network dynamics. In this case, real-valued species concentration variables are updated by approximating a solution vector in respect to the current timestep. In a simple network, the rate of change for each species concentration is equated to concentration levels of other interacting species concentrations. Each interaction has a positive or negative effect on the rate of concentration change for a given species. Equation 2.25 mathematically defines this interaction among species.

$$\frac{dX_i}{dt} = V_p^i(X) - V_c^i(X) \quad (2.25)$$

where X_i is the vector of species concentrations, V_p^i and V_c^i are the kinetic rate laws previously mentioned in the section 2.3.4 that produce and consume X_i respectively.

The main advantage of the rate equation approach is that dynamics are totally characterized by a system of ODEs. Accordingly, the modeler can readily analyze and predict model behavior by applying numerical analysis methods to the system of ODEs. For example, stability of steady state points can be determined by calculating the eigenvalues of the Jacobian matrix of the nonlinear system. The main disadvantage for this method is that it is totally deterministic. Consequently, this approach ignores the inherent randomness or ‘noise’ that acts upon biological systems. However, ‘noise’ can be added to this deterministic model by augmenting it with stochastic terms.

Unlike the deterministic rate equation approach, the stochastic kinetics approach that incorporates the highest level of detail explicitly takes into account the randomness or fluctuations in rates of gene expression. Results from Arkin et al. suggest that such random behavior “can produce a highly erratic time pattern of protein production in each individual cell and a wide diversity of protein concentrations across a cell population at any instant in time [3:1633].” Arkin et al. state the importance of considering stochastic behavior: “When two independently produced regulatory proteins acting at low cellular concentrations competitively control a switch point in a pathway, stochastic variations in their concentrations can produce probabilistic pathway selection.... [3:1633]” In short, the stochastic kinetics approach attempts to closely model important cellular events that are inherently random within a genetic regulatory network. This method enables the modeler to track events at the micro level. Detailed information on the stochastic approach can be found in [3:1633–1648].

Likewise, this approach has a significant advantage and major disadvantage. “[It] is impressively complete and yields a detailed picture of the behavior of the system modeled.” The major disadvantage is that model results “comes at a high

computational cost and sacrifices any immediate prospect of analytical treatment [10:270].”

2.9 *Gepasi*

“Gepasi (version 3.21) is a software package for biochemical systems. It simulates the kinetics of systems of biochemical reactions and provides a number of tools to fit models to data, optimize any function of the model, perform metabolic control analysis, and linear stability analysis [17:1]”. Gepasi uses the chemical-reaction-centered approach in building the computational model. In this mode, the user enters a system of reaction equations using standard chemical notation. From this user input, the program automatically builds and solves the system of ODEs. Unfortunately, the user is unable to view the system of ODEs generated by the program.

After building the model, the user can perform two kinds of simulations—time course and steady state. Time course simulations depict how kinetics of the reaction network evolve over time from a starting point. In setting up this type of simulation, the user defines the end time, number of sampling points, and output configuration for the resultant data file. Output files are configured in such a way that third party applications can readily import Gepasi data files [17:1].

Steady state simulations attempt to find a steady state near a starting point. Gepasi cannot find some types of steady states (e.g., saddle points). Steady state results are sent to a data file. These files contain calculated steady state concentration for each metabolite via a row vector [17].

Additional features listed in the Gepasi user manual are ‘scanning parameter space’ and ‘fitting model parameters to experimental data’. In the scanning parameter mode, the program automatically performs a sensitivity analysis on model parameters as specified by the user. In the fitting mode, the program minimizes the sum of squares of residuals between predicted values from the model and experimental data [17].

2.10 *Virtual Cell*

Virtual Cell is another software package for modeling cellular biological processes. “It is based on the mapping of experimental biochemical and electrophysiological data onto experimental images. The framework is designed to enable the construction of complex general models that encompass the general class of problems coupling reaction and diffusion [23:228].” Unlike Gepasi, a graphical approach is incorporated into Virtual Cell. In this approach, the user constructs a model by placing and manipulating abstract modeling objects onto an active workspace. Modeling objects can be edited, viewed, stored in a remote database, and analyzed using the WWW-based user interface [23:230].

For building models and running simulations, the user interacts with the Virtual Cell application through a GUI enabling access to various editors. After running the simulation, the user can view results (in graphical and tabular format) in separate windows. Program output includes ODEs, graphical display of simulation results, and simulation data in tabular format. The user has several choices of formats in exporting the data. Additional information about Virtual Cell and access to its WWW-based user interface is at <http://www.nrcam.uchc.edu/vcellR3/login/tutorial.jsp>.

2.11 *E-CELL2*

E-CELL is a software package that enables the user to model metabolic pathways and higher-order cellular processes, such as protein synthesis and signal transduction [28]. This package employs the ‘Substance-Reactor Model’ in depicting the structure of a cell and the chemical reactions that take place within a cell. In this fashion, three classes of objects are used to model cell activity: substances, reactors, and systems. Reactors calculate the changes in the amount of substances over time. Systems express the location of reactors and substances. Reactors implement the specified reaction mechanism (e.g., Michaelis-Menten rate law) [15].

The user can view simulation results within several windows. The state of a simulation run can be captured and subsequently saved. The user is able to save simulation results and whole cell state information of the model at a given time to a file. Contents in the data file include simulation data on all substances and reactors, time, quantity, an average quantity, the maximum quantity, the minimum quantity, the average concentration, the maximum concentration, and the minimum concentration [15]. Additional information can be accessed from [28] and [15].

2.12 *BioSPICE*

The Defense Advanced Research Projects Agency's (DARPA) Bio-Computation program is aimed at exploring and developing computational methods and models at the biomolecular and cellular levels. This organization released a software bundle called **BioSPICE** (version 1.1.). It contains an assortment of biochemical modeling tools. Software packages included in this bundle are—among others—Grass, Simpathica, Pathway Builder, Charon, Biosketchpad, and JigCell.

Charon and Biosketchpad are used in conjunction with each other, forming BioCharon. This application incorporates a *graphical approach*. In contrast, Jigcell incorporates a *spread sheet* approach. I describe BioCharon and JigCell in chapter III. I list and analyze simulation results produced by these software tools in chapter IV.

III. Intracellular Modeling

3.1 Overview

The intracellular environment involves intricate dynamics. Understanding the underlying logic that controls these dynamics is the foundation for illuminating overall cell functionality. Despite the emergence of powerful techniques in sequencing genes and profiling RNA and protein at the genomic level, “there is a considerable gap between the availability of new sequence data and a scientific understanding of that information [9:3].” Mathematical modeling can help bridge that gap between the large repository of data and the scientific understanding of that information.

As previously mentioned in section 2.1, formulating a computational model involves a series of detailed steps that quickly overwhelm the pencil-and-paper approach. Consequently, there is a bona fide need for the development of software tools that ‘assist’ the modeler in designing and implementing computational models of cellular networks. Two such software packages are discussed in this thesis—BioCharon and JigCell.

In this chapter, I mainly focus on the modeling of non-biological and naturally occurring networks. I furnish background information on the non-biological network—the *Brusselator*—and naturally occurring networks—the lysis-lysogeny pathway of a mutant bacteriophage, the control system that activates bioluminescence in *Vibrio fischeri* (*V. fischeri*), molecular control of the cell cycle in budding yeast, and a glycolytic pathway. I also give a brief description on both software packages to include system configuration, system capabilities, user interaction, and program output.

Besides describing each software package and cellular network, I lay the foundation of how computational models are derived from cellular wiring diagrams. I incorporate the rate-reaction approach in each derivation. Each cellular network model in this thesis is converted into MATLAB code. Simulation results obtained

from MATLAB are regarded as baseline measurements for comparison to simulation results produced by BioCharon and JigCell.

3.2 Rate-Equation Approach

As previously mentioned in section 2.8, mathematical modeling involves three common approaches: Boolean, rate-equation, and stochastic. The rate-equation approach—the only approach that is implemented in this thesis—involves three main steps.

1. Convert a wiring diagram into a set of reaction equations.
2. Derive individual reaction rate laws (i.e., reaction velocities) from the set of reaction equations derived in step 1.
3. Express each species (state variable) with the appropriate subset of reaction rate velocities derived in step 2.

Rules to derive individual reaction rate laws (step 2) can be found in [25:8–10] and [12:11–13]. The v_i 's in equations 3.3 and 3.4 are derived using equation 2.7 (just as equation 2.15 results from equation 2.7).

As a concrete example, I implement the rate-equation approach for the following reaction system illustrated in figure 3.1, using steps one through three and basic laws of kinetics (equations 2.6 and 2.7). This reaction system is published in [24:397].

Step 1a:



where k_1 is the forward reaction rate constant and k_{-1} is the reverse reaction rate constant for equation 3.1.

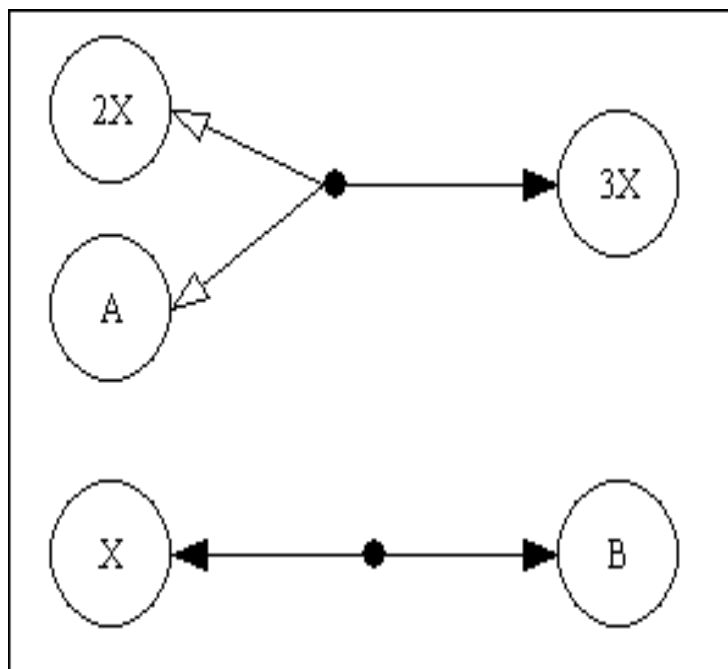


Figure 3.1. This is a possible wiring diagram for the system of reaction equations proposed by [24:397].

Step 1b:



where k_2 is the forward reaction rate constant and k_{-2} is the reverse reaction rate constant for equation 3.2.

Step 2:

$$v_1 = -k_1[X]^2[A] + k_{-1}[X]^3 \quad (3.3)$$

$$v_2 = -k_2[X] + k_{-2}[B] \quad (3.4)$$

Step 3a:

$$\begin{aligned} \frac{d[X]}{dt} &= 2v_1 - 3v_1 + v_2 \\ &= -v_1 + v_2 \\ &= k_1[X]^2[A] - k_{-1}[X]^3 - k_2[X] + k_{-2}[B] \end{aligned}$$

Step 3b:

$$\begin{aligned}\frac{d[A]}{dt} &= v_1 \\ &= -k_1[X]^2[A] + k_{-1}[X]^3\end{aligned}$$

Step 3c:

$$\begin{aligned}\frac{d[B]}{dt} &= -v_2 \\ &= k_2[X] - k_{-2}[B]\end{aligned}$$

Upon inspection, one notices that $\frac{d[X]}{dt} + \frac{d[A]}{dt} + \frac{d[B]}{dt} = 0$. This implies that

$$([X] + [A] + [B]) = T_1. \quad (3.5)$$

where T_1 is a constant. Equation 3.5 defines a conservation relation among three species. Even though concentrations of these species may change over time, the sum of their concentrations remain constant.

Equation 3.5 can also be rewritten as equation 3.6.

$$[B] = T_1 - ([A] + [X]). \quad (3.6)$$

Consequently, the dynamics of this reaction system can be expressed by a system of two non-linear ODEs and one algebraic equation. This system is defined by equation 3.7.

$$\begin{aligned}\frac{d[X]}{dt} &= k_1[X]^2[A] - k_{-1}[X]^3 - k_2[X] + k_{-2}[B] \\ \frac{d[A]}{dt} &= -k_1[X]^2[A] + k_{-1}[X]^3 \\ [B] &= T_1 - ([X] + [A])\end{aligned}$$

3.3 BioCharon

BioCharon is a software package for designing, simulating, and analyzing complex biomolecular networks using hybrid systems (a hybrid system involves both discrete and continuous dynamics) [5:1]. It implements a graphical approach for user interaction in building biomolecular models. This package is comprised of two software programs—Bio Sketch Pad and Charon. Bio Sketch Pad is a GUI-based drawing program and Charon is a hybrid-systems simulation program [5:1].

The user designs and simulates a biomolecular model by completing the following steps.

1. Input model into Bio Sketch Pad (model definition).
2. Translate model into Charon programming language.
3. Load translated model file (*modelname.prj*) into Charon Visual Simulator.
4. Compile loaded model file (select *Generate Simulator* command).
5. Select simulation options (*number of steps* and *integration steps* [i.e., integration step size]).
6. Select simulation type (*background simulation* or *display simulation trace*).
7. Start simulation (assert *start* command button).

Figure 3.2 sketches out these steps.

3.3.1 Bio Sketch Pad. Bio Sketch Pad (BSP) is an interactive tool for modeling and designing biomolecular and cellular networks using a graphical front end [5:2]. This front end (see figure 3.3) implements the rate-equation approach via graphical interaction with the user. In this manner, the modeler defines a model in the GUI by assembling together defined nodes and arcs. Additionally, BSP automatically checks for syntax errors while the user is building the model.

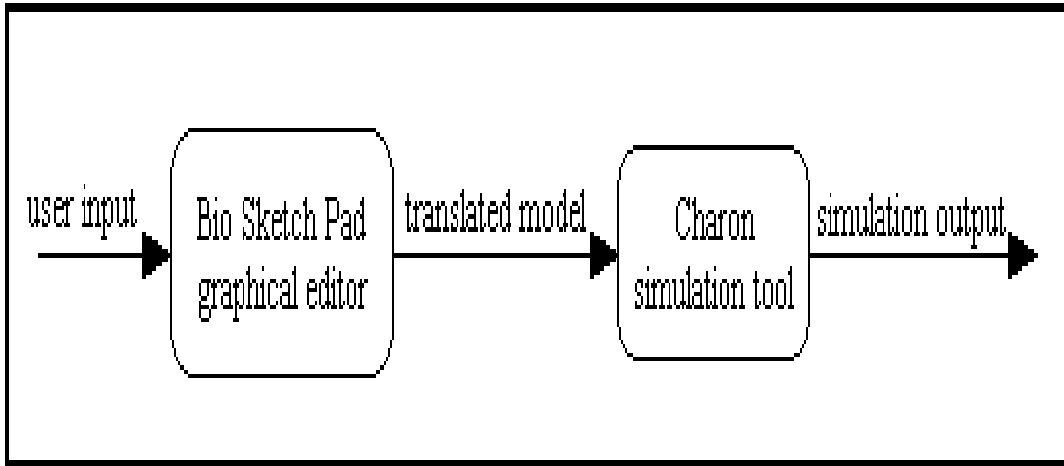


Figure 3.2. This depicts the process of how a computational model is generated from user input via BioCharon. The user first designs a model in BSP's GUI. Then, this completed model is translated into the Charon programming language. Finally, this translated model is loaded into Charon's visual simulator for subsequent simulations.

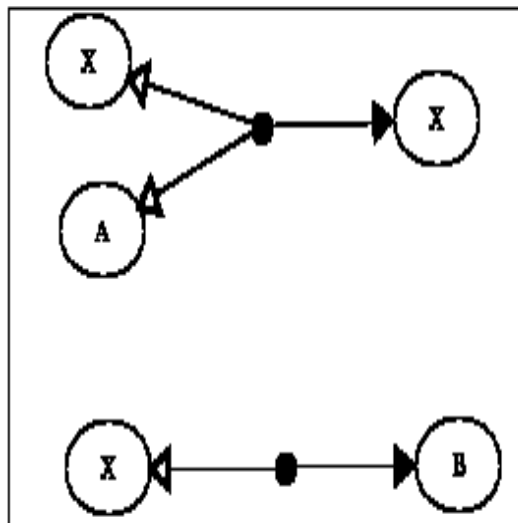


Figure 3.3. This is an entry into BSP that implements the graphical approach for user interaction.

For example, BSP assures that nodes and arcs—main elements of BSP—are properly oriented. Consequently, the user can only connect species to reactions and vice versa. BSP cues the user by *coloring* intended coupling points. Green signifies that the user is attempting a legal operation. Contrarily, red means that the user is attempting an illegal operation.

The orientation and parameter definitions of nodes and arcs completely characterize the model. The nodes are either species, chemical reactions, or regulations. The arcs describe relations or interactions between the nodes [5:2].

3.3.2 Charon. Charon is a hybrid-system analysis and simulation tool [5:1]. After loading translated model files from BSP into the Charon visual simulator, the modeler can perform simulations and analyze the results within the visual simulator’s GUI. The process of how a graphical model created in BSP is translated to Charon programming code is outlined in figure 3.2.

Charon’s programming language serves as an interface between BSP and Charon. Since the modeler may not be able to directly implement some desired characteristics of a cellular network using BSP as a front-end (e.g., stochastic behavior, inhibition, or Michaelis-Menten rate law), it may be necessary for the modeler to include some functions directly into the Charon programming code. Additional information about BSP and Charon is located in [5].

3.4 JigCell

JigCell is another software package that can be used for modeling cellular networks. It is comprised of three main software programs: JigCell Model Builder (JCMB), JigCell Run Manager, and Comparator. The user designs and modifies models with JCMB. The Run Manager is the platform for running simulations of models created in JCMB. The user is able to visually verify the ‘goodness’ of a model using the Comparator. This program generates a combined plot of model predictions

and experimental temporal data points for direct comparison. Additional information about JigCell is located in [30].

3.5 *Biomolecular networks*

“Technological advancements combined with intensive DNA sequencing efforts have generated an enormous database of sequence information over the past decade [21:153].” Mathematical modeling is used to help bridge the gap between this huge database of sequence information and the scientific understanding of it. Two types of biomolecular networks are generally modeled for the enhancement of this understanding—naturally occurring and abstract networks. Naturally occurring networks are specific biological systems, such as *E. Coli* and *M. tuberculosis* (prokaryotes) and *Saccharomyces cervisiae* and *Homo sapiens* (eukaryotes). Abstract networks are synthetic networks. In the remaining sections of this chapter, I will give background information on naturally occurring and abstract models as well as derive computational models from each network wiring diagram.

3.6 *Non-Biological Network: Modified Brusselator*

“The accurate mathematical description of synthetic networks provides the foundation for describing complex, naturally occurring networks [10:268–269].” Insight gained from synthetic networks can sometimes be applied to a class of biological subsystems. The Brusselator is a synthetic network that is covered in this section. “It is well known that the Brusselator model reveals simple oscillations and damped oscillations by the suitable parameters [18:249].”

First proposed by Prigogine and Lefever, the Brusselator model is represented by the following system of reaction equations [22:1697].





The overall reaction is $A+B \rightarrow E+D$ [22:1697]. The concentrations of A , B , D , and E are assumed to remain constant and greater than zero. All forward kinetic rate constants are set to one. This system only involves two intermediate components (species)— X and Y [22:1697]. X changes Y into X by the autocatalytic reaction in equation 3.9 [18:249]. The reaction rate for that equation is increased as the product forms. Furthermore, the stoichiometric coefficients for X in that equation are two and three on the substrate side and product side, respectively.

Morikawa et al. generalized equation 3.9. They substituted $(2 + \epsilon)$ for the stoichiometric coefficient for X on the product side of that equation. The original system of reaction equations is modified by replacing equation 3.9 with equation 3.11.



The free parameter ϵ represents the *strength* of the autocatalytic process [18:249]. When

ϵ is set to one, equation 3.11 reduces to equation 3.9. I postulate that as the value of ϵ is increased, the steady state concentration of species X is also increased. Using steps two and three in section 3.2, the computational model is derived from the following equations.

$$v_1 = -[A] \quad (3.12)$$

$$v_2 = -[B][X] \quad (3.13)$$

$$v_3 = -[X]^2[Y] \quad (3.14)$$

$$v_4 = -[X] \quad (3.15)$$

Reaction velocities v_1 through v_4 are collectively used to express the dynamics (fluxes) of species X and Y . Model components A , B , D , and E are not regarded as species or state variables of the computational model, because their concentration levels remain constant. The fluxes of species X and Y are characterized by the following system of ODEs.

$$\begin{aligned} \frac{d[X]}{dt} &= -v_1 + v_2 + 2v_3 - (2 + \epsilon)v_3 + v_4 \\ &= -v_1 + v_2 + 2v_3 - 2v_3 - \epsilon v_3 + v_4 \\ &= -v_1 + v_2 - \epsilon v_3 + v_4 \\ &= [A] - [B][X] + \epsilon[X]^2[Y] - [X] \\ &= [A] - ([B] + 1)[X] + \epsilon[X]^2[Y] \end{aligned} \quad (3.16)$$

$$\begin{aligned} \frac{d[Y]}{dt} &= -v_2 + v_3 \\ &= [B][X] - [X]^2[Y] \end{aligned} \quad (3.17)$$

By setting $[A] = a$, $[B] = b$, $[X] = x$, and $[Y] = y$, equations 3.16 and 3.17 are equivalent to the following system of equations.

$$\frac{dx}{dt} = a - (b + 1)x + \epsilon x^2 y \quad (3.18)$$

$$\frac{dy}{dt} = bx - x^2 y \quad (3.19)$$

Equations 3.18 and 3.19 can be rewritten as $\frac{dx}{dt} = f_1$ and $\frac{dy}{dt} = f_2$, respectively. Functions f_1 and f_2 are defined by equations 3.20 and 3.21.

$$f_1 = a - (b + 1)x + \epsilon x^2 y \quad (3.20)$$

$$f_2 = bx - x^2 y \quad (3.21)$$

The critical point solution for the system of equations 3.16 and 3.17 is derived by setting $f_1 = 0$ and $f_2 = 0$.

$$a - (b + 1)x + \epsilon x^2 y = 0 \quad (3.22)$$

$$bx - x^2 y = 0 \quad (3.23)$$

Solving for y in 3.23 results in equation 3.24. Inserting equation 3.24 into equation 3.22 and then solving for x results in equation 3.25.

$$y = \frac{b}{x} \quad (3.24)$$

$$x = \frac{a}{b(1 - \epsilon) + 1} \quad (3.25)$$

Equations 3.24 and 3.25 coupled together imply $y = b \left(\frac{b(1-\epsilon)+1}{a} \right)$. Therefore the critical point solution is represented by equation 3.26.

$$(\hat{x}, \hat{y}) = \left(\frac{a}{b(1 - \epsilon) + 1}, b \left(\frac{b(1 - \epsilon) + 1}{a} \right) \right) \quad (3.26)$$

From equation 3.26, it is clear that the initial concentration values for species X and Y have no impact on the critical point solution. It is solely a function of controllable parameters a , b , and ϵ .

In stability analysis, time developments of the fluctuations (x, y) around the critical point solution $(y_{cp} \equiv (\hat{x}, \hat{y}))$ are approximated by the following linearized equations [18:250].

$$\begin{aligned}\frac{dx}{dt} &= L_{11}x + L_{12}y \\ \frac{dy}{dt} &= L_{21}x + L_{22}y\end{aligned}$$

L_{11} , L_{12} , L_{21} , and L_{22} are elements of the Jacobian matrix (J). The constant values for these elements are derived from the following set of equations.

$$\begin{aligned}L_{11} &= \frac{\partial f_1(\mathbf{y}_{cp})}{\partial x} = (2\epsilon - 1)b - 1 \\ L_{12} &= \frac{\partial f_1(\mathbf{y}_{cp})}{\partial y} = \epsilon\hat{x}^2 \\ L_{21} &= \frac{\partial f_2(\mathbf{y}_{cp})}{\partial x} = -b \\ L_{22} &= \frac{\partial f_2(\mathbf{y}_{cp})}{\partial y} = -\hat{x}^2\end{aligned}$$

These elements are used to populate the following Jacobian matrix.

$$\begin{pmatrix} (2\epsilon - 1)b - 1 & \epsilon\hat{x}^2 \\ -b & -\hat{x}^2 \end{pmatrix} \quad (3.27)$$

At this point, Morikawa et al. ([18:250]) set

$$b = \frac{1}{(2\epsilon - 1)}. \quad (3.28)$$

Equation 3.28 implies $\epsilon > 1/2$, because $b > 0$. However, the value of ϵ must be an integer equal to or greater than one in order to maintain physical relevance. Subsequent substitutions for b imply $L_{11} = 0$ and $L_{21} = \frac{-1}{(2\epsilon-1)}$. Equation 3.27

reduces to the new Jacobian matrix listed in equation 3.29.

$$\begin{pmatrix} 0 & \epsilon \hat{x}^2 \\ \frac{-1}{(2\epsilon-1)} & -\hat{x}^2 \end{pmatrix} \quad (3.29)$$

The trace [$tr(J)$], determinant [$det(J)$], and characteristic equation [$char(J)$] are derived from equation 3.29 as follows.

$$\begin{aligned} tr(J) &= L_{11} + L_{22} \\ &= 0 + -\hat{x}^2 \\ &= -\hat{x}^2 \end{aligned} \quad (3.30)$$

$$\begin{aligned} det(J) &= L_{11}L_{22} - L_{12}L_{21} \\ &= 0 + \epsilon \hat{x}^2 \left(\frac{1}{(2\epsilon-1)} \right) \\ &= \frac{\epsilon \hat{x}^2}{(2\epsilon-1)} \end{aligned} \quad (3.31)$$

$$\begin{aligned} char(J) &= \lambda^2 - tr(J)\lambda + det(J) \\ &= \lambda^2 + \hat{x}^2\lambda + \frac{\epsilon \hat{x}^2}{(2\epsilon-1)} \end{aligned} \quad (3.32)$$

I use the quadratic formula $\left(\frac{-\hat{b} \pm \sqrt{(\hat{b}^2 - 4\hat{a}\hat{c})}}{2\hat{a}} \right)$ to solve for eigenvalues $\lambda_{1,2}$ by setting equation 3.32 to zero. In this case $\hat{a} = 1$, $-\hat{b} = tr(J)$, and $\hat{c} = det(J)$.

$$\begin{aligned} \lambda_{1,2} &= \frac{tr(J) \pm \sqrt{(-tr(J))^2 - 4(det(J))}}{2} \\ &= \frac{-\hat{x}^2 \pm \sqrt{(\hat{x}^2)^2 - 4\left(\frac{\epsilon \hat{x}^2}{(2\epsilon-1)}\right)}}{2} \\ &= \frac{-\hat{x}^2 \pm \sqrt{\hat{x}^4 - 4\left(\frac{\epsilon \hat{x}^2}{(2\epsilon-1)}\right)}}{2} \end{aligned} \quad (3.33)$$

An observation from equation 3.33 is that $-\hat{x}^2 < 0$ (assuming $\hat{x} \neq 0$). If the expression $\left[\hat{x}^4 - 4\left(\frac{\epsilon \hat{x}^2}{(2\epsilon-1)}\right) \right] < 0$ in equation 3.33 is true, then the values of

$\lambda_{1,2}$ contain negative real parts and imaginary parts. As a result, the critical point solution is stable. This expression can be solved for \hat{x} ; however, it is desirable to solve for a controllable parameter, such as a . Below, I derive a condition that ensures that the values of $\lambda_{1,2}$ contain imaginary parts.

$$\begin{aligned}
\hat{x}^4 - 4 \left(\frac{\epsilon \hat{x}^2}{(2\epsilon - 1)} \right) &< 0 \\
\hat{x}^2 - 4 \left(\frac{\epsilon}{(2\epsilon - 1)} \right) &< 0 \\
\frac{(2\epsilon - 1)^2 (a)^2}{\epsilon^2} - \frac{4\epsilon}{2\epsilon - 1} &< 0 \\
\frac{(2\epsilon - 1)^2 (a)^2}{\epsilon^2} &< \frac{4\epsilon}{2\epsilon - 1} \\
a^2 &< \left(\frac{4\epsilon}{2\epsilon - 1} \right) \left(\frac{\epsilon^2}{(2\epsilon - 1)^2} \right) \\
a^2 &< \frac{4\epsilon^3}{(2\epsilon - 1)^3} \\
a &< 2 \left(\frac{\epsilon}{2\epsilon - 1} \right)^{\frac{3}{2}}
\end{aligned} \tag{3.34}$$

$$\tag{3.35}$$

Equation 3.34 is derived by making a substitution for x^2 (squaring both sides of equation 3.25). The critical point solution for the system of equations 3.16 and 3.17 is stable when conditions defined by equations 3.28 and 3.35 and ϵ is an integer equal to or greater than one are satisfied.

3.7 Natural Networks

“Although abstract models can offer insight into basic mechanisms, modeling must ultimately be connected to specific systems so that verifiable predictions can be made [10:269].” Since there is a large reservoir of experimental data currently available on a diverse set of prokaryotes and eukaryotes [26:247], the stage is set to enable tight coupling of candidate models to these naturally occurring systems. In this section, three natural systems are considered for modeling in JigCell and

BioCharon—bacteriophage, *Vibrio fischeri*, and budding yeast (*Sachharomyces cerevisiae*).

3.7.1 Lysis-Lysogeny Pathway. Hasty et al. propose a system of reaction equations to characterize the lysis-lysogeny pathway of a mutant λ phage. Unlike a wild type λ phage, the pathway of a mutant λ phage consists of two (instead of three) operator sites—OR2 and OR3. This “system is a plasmid consisting of the $P_R - P_{RM}$ operator region and components necessary for transcription, translation, and degradation [11:2076].”

The basic dynamical properties of this pathway are as follows. “The gene *cI* expresses repressor (CI), which in turn dimerizes and binds to the DNA as a transcription factor..... Binding at *OR2* enhances transcription, which takes place downstream of *OR3*, whereas binding at *OR3* represses transcription, effectively turning off production [11:2076].” The system of reaction equations that describe flux of the lysis-lysogeny pathway of a mutant λ virus is described by equations 3.36–3.42. X , X_2 , and D denote the concentrations of repressor (a protein), repressor dimer, and DNA promoter sites, respectively. DX_2 , DX_2^* , and DX_2X_2 denote binding to sites OR2, OR3, or both, respectively. P denotes the concentration of RNA polymerase, and n is the number of proteins per mRNA transcript [11:2076].



where r is the basal rate.



where k_1 is the forward reaction rate and k_{-1} is the reverse reaction rate.



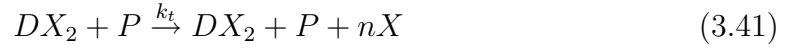
where k_2 is the forward reaction rate and k_{-2} is the reverse reaction rate.



where k_3 is the forward reaction rate and k_{-3} is the reverse reaction rate.



where k_4 is the forward reaction rate and k_{-4} is the reverse reaction rate.



where k_t is the reaction rate.



where k_d is the reaction rate.

The following reaction velocities are calculated from the previously defined system of reaction equations.

$$\begin{aligned} v_1 &= r \\ v_2 &= -k_1[X]^2 + k_{-1}[X_2] \\ v_3 &= -k_2[D][X_2] + k_{-2}[DX_2] \\ v_4 &= -k_3[D][X_2] + k_{-3}[DX_2^*] \\ v_5 &= -k_4[DX_2][X_2] + k_{-4}[DX_2X_2] \\ v_6 &= -k_t[DX_2][P] \\ v_7 &= -k_d[X] \end{aligned}$$

A system of ODEs—expressing the flux of each species—can now be formulated from this set of reaction velocities.

$$\frac{d[X]}{dt} = v_1 + 2v_2 - nv_6 + v_7 \quad (3.43)$$

$$\frac{d[X_2]}{dt} = -v_2 + v_3 + v_4 + v_5 \quad (3.44)$$

$$\frac{d[D]}{dt} = v_3 + v_4 \quad (3.45)$$

$$\frac{d[DX_2]}{dt} = -v_3 + v_5 + v_6 - v_6 \quad (3.46)$$

$$\frac{d[DX_2^*]}{dt} = -v_4 \quad (3.47)$$

$$\frac{d[DX_2X_2]}{dt} = -v_5 \quad (3.48)$$

Since $\frac{d[D]}{dt} + \frac{d[DX_2]}{dt} + \frac{d[DX_2^*]}{dt} + \frac{d[DX_2X_2]}{dt} = 0$ (from equations 3.45 - 3.48), there exists a constant relation among the concentrations of D , DX_2 , DX_2^* , and DX_2X_2 promoter sites. At any time, the **total** concentration—denoted by d_T —remains constant among these four promoter sites. The value of d_T is determined by the sum of initial concentrations from all four promoter sites. Equation 3.49 defines that constant relation.

$$d_T = [D] + [DX_2] + [DX_2^*] + [DX_2X_2] \quad (3.49)$$

The above system of ODEs reduce to the following computational model.

$$\frac{d[X]}{dt} = r + 2(k_{-1}[X_2] - k_1[X]^2) + nk_t[DX_2][P] - k_d[X] \quad (3.50)$$

$$\begin{aligned} \frac{d[X_2]}{dt} = & k_1[X]^2 - k_{-1}[X_2] - k_2[D][X_2] + k_{-2}[DX_2] - k_3[D][X_2] + \\ & k_{-3}[DX_2^*] - k_4[DX_2][X_2] + k_{-4}[DX_2X_2] \end{aligned} \quad (3.51)$$

$$\frac{d[D]}{dt} = k_{-2}[DX_2] - k_2[D][X_2] - k_3[D][X_2] + k_{-3}[DX_2^*] \quad (3.52)$$

$$\frac{d[DX_2]}{dt} = k_{-4}[DX_2X_2] - (k_{-2}[DX_2] - k_2[D][X_2]) - k_4[DX_2][X_2] \quad (3.53)$$

$$\frac{d[DX_2^*]}{dt} = k_3[D][X_2] - k_{-3}[DX_2^*] \quad (3.54)$$

$$\frac{d[DX_2X_2]}{dt} = k_4[DX_2][X_2] - k_{-4}[DX_2X_2] \quad (3.55)$$

Solving for $[DX_2X_2]$ in equation 3.49, a substitution is made for $[DX_2X_2]$ in equations 3.51 and 3.53. Equation 3.51 reduces to equation 3.56, and equation 3.53 reduces to equation 3.57.

$$\begin{aligned} \frac{d[X_2]}{dt} = & k_1[X]^2 - k_{-1}[X_2] - k_2[D][X_2] + k_{-2}[DX_2] - k_3[D][X_2] + \\ & k_{-3}[DX_2^*] - k_4[DX_2][X_2] + k_{-4}(d_T - ([D] + [DX_2] + [DX_2^*])) \end{aligned} \quad (3.56)$$

$$\begin{aligned} \frac{d[DX_2]}{dt} = & k_{-4}(d_T - ([D] + [DX_2] + [DX_2^*])) - (k_{-2}[DX_2] - k_2[D][X_2]) - \\ & k_4[DX_2][X_2] \end{aligned} \quad (3.57)$$

As a result, the computational model can now be expressed by a system of five ODEs (equations 3.50, 3.56, 3.52, 3.57, and 3.54). This reduced system has a unique critical point solution.

Hasty et al. set $x = [X]$, $y = [X_2]$, $d = [D]$, $u = [DX_2]$, $v = [DX_2^*]$, and $z = [DX_2X_2]$ [11:2076]. Then, Hasty et al. reduce this system of six equations (3.60 - 3.55) into two equations by deriving the following system of equations:

$$\begin{aligned} y &= K_1x^2 \\ u &= K_1k_2dx^2 \\ v &= \sigma_1K_1K_2dx^2 \\ z &= \sigma_2(K_1K_2)^2dx^4 \end{aligned} \quad (3.58)$$

where $K_1 = \frac{k_1}{k_{-1}}$, $K_2 = \frac{k_2}{k_{-2}}$, $\sigma_1 = k_3/k_2$, and $\sigma_2 = k_4/k_2$.

Also, the total concentration of DNA promoter sites d_T is constant, so that $d_T = d + u + v + z$. I rewrite this equation with respect to d , so that the entire

system has only one degree of freedom.

$$d = \frac{d_T}{k_1 k_2 x^2 + \sigma_1 k_1 k_2 (x)^2 + \sigma_2 (k_1 k_2)^2 (x)^4} \quad (3.59)$$

Finally, Hasty et al. presents the nondimensional equation that represents the dynamics of the entire system.

$$x' = \frac{\alpha x^2}{1 + (1 + \sigma_1)x^2 + \sigma_2 x^4} - \gamma x + 1 \quad (3.60)$$

where $\alpha \equiv nk_t p_0 d_T / r$ and $\gamma = k_d / (r \sqrt{k_1 k_2})$.

The critical points for 3.60 satisfy the roots for the fifth degree polynomial listed in equation 3.61. This equation is obtained by setting the right hand side of equation 3.60 to zero and performing algebraic manipulations on the resulting equation.

$$\gamma \sigma_2 x^5 - \sigma_2 x^4 + \gamma x^3 (1 + \sigma_1) - x^2 (\alpha + 1 + \sigma_1) + \gamma x - 1 = 0 \quad (3.61)$$

Five roots can be determined from equation 3.61. This is discussed further in section 4.4.1.

3.7.2 Vibrio fischeri. As previously discussed in section 3.4, *V. fischeri* has the ability of bioluminescence. “The transcriptional activation of the *lux* genes in the bacterium controls this luminescence [1:8]”. The set of *lux* genes that controls luminescence is regarded as an operon.

An operon is a sequence of closely associated genes that regulate enzyme production. An operon includes one or more *structural genes*, which carry information for the synthesis of specific proteins such as enzyme molecules, and *regulatory sites*, which control the expression of the structural genes. A regulator gene works in conjunction with the operon but may be located some distance from it [6:180].

The lux regulon (i.e., lux genes associated with regulation) is organized into two transcriptional units: O_L and O_R . O_L contains the *luxR* gene and O_R contains seven genes (*luxICDABEG*) [1:8]. These structural genes encode the following proteins.

luxR encodes the protein LuxR [1:8].

luxI “produces the protein LuxI, which is required for endogenous production of the autoinducer Ai, a small membrane-permeable signal molecule [1:8].”

luxA and luxB are templates for luciferase subunits [1:8].

luxC and luxD, and luxE “code for proteins of the fatty acid reductase, which generates aldehyde substrate for luciferase [1:8].”

luxG “is thought to encode a flavin reductase [1:8].”

The architecture of the lux regulon is illustrated in Figure 3.4. Additionally, that figure contains a description of species interactions—affecting luminescence—that takes place within the cell. The pathway description is extracted from [1:8].

Due to the advanced nature of deriving the computational model for characterizing luminescence of *V. fischeri*, I omit steps 1-3 as described in section 3.2. The complete set of reaction equations is listed in appendix A as input for a JigCell calculation. In addition, a resultant rate law (i.e., v_i) is automatically generated for each reaction equation entry during model formulation. The following computational model for luminescence is submitted by Alur et al. [1:8]. Descriptions of variables and constants for this model are listed in tables 3.1 and 3.2, respectively.

$$\dot{x}_0 = k_G x_0$$

$$\dot{x}_1 = T_c(\psi(x_8, K_{Co-icdabeg}\Phi(c_{CRP}, K_{CRPr}, V_{CRPr}) + b) - \frac{x_1}{H_{RNA}} - k_G x_1$$

$$\dot{x}_2 = T_c(\Phi(x_8, K_{Co-icdabeg}\psi(c_{CRP}, K_{CRPr}, V_{CRPr}) + b) - \frac{x_2}{H_{RNA}} - k_G x_2$$

$$\dot{x}_3 = T_l x_1 - x_3/H_{sp} - r_{AIR} x_7 x_3 + r_{Co} x_8 - k_G x_3$$

$$\dot{x}_4 = T_l x_2 - x_4/H_{sp} - k_G x_4$$

$$\begin{aligned}
\dot{x}_5 &= T_l x_2 - x_5/H_{sp} - k_G x_5 \\
\dot{x}_6 &= T_l x_2 - x_6/H_{sp} - k_G x_6 \\
\dot{x}_7 &= x_0(r_{AI}x_4 - r_{AIR}x_7x_3 + r_{Co}x_8) - x_7/H_{AI} \\
\dot{x}_8 &= r_{AIR}x_7x_3 - x_8/H_{sp} - r_{Co}x_8 - k_G x_8
\end{aligned} \tag{3.62}$$

where

$$k_G = k_g(1 - x_0/x_{0max}) \tag{3.63}$$

$$\Phi(X, K_{Xm}, V_{Xm}) = \frac{X^{V_{Xm}}}{K_{Xm}^{V_{Xm}} + X^{V_{Xm}}} \tag{3.64}$$

$$\psi(X, K_{Xm}, V_{Xm}) = 1 - \Phi(X, K_{Xm}, V_{Xm}) \tag{3.65}$$

Φ is a sigmoid function that serves as a switch for regulating (from low to high activation levels) the transcription of mRNA. As the concentration of the regulatory species X increases, the value of Φ increases. The maximum value of Φ is bounded by the limit of Φ , because Φ is monotone. In contrast, the ψ function has the opposite effect of the sigmoid function. As the concentration of the regulatory species X increases, the value of ψ decreases. The minimum value is bounded by the limit of ψ (usually zero).

Table 3.1. Species Cross Reference List (*V. fischeri*) [1:8]

Species	Description
x_0	scaled population (population x v_b/V)
x_1	luxR (mRNA transcribed from O_L)
x_2	luxICDABEG (mRNA transcribed from O_R)
x_3	protein LuxR
x_4	protein LuxI
x_5	protein LuxA/B ([LuxA/B] reflects cell luminescence)
x_6	protein LuxC/D/E
x_7	autoinducer Ai
x_8	complex Co

Table 3.2. Description of Model Parameters (*V. fischeri*) [1:8]

Parameter	Description
T_c	Maximum transcription rate
T_l	Maximum translation rate
H_{RNA}	RNA half-life
H_{sp}	Stable protein half-life
H_{up}	Unstable protein half-life
H_{AI}	Ai half-life
r_{AII}	Rate constant at which LuxI makes Ai
r_{AIR}	Rate constant Ai binding to LUXR
r_{Co}	Rate constant of Co dissociation
v_{CRPr}	Cooperatively coefficient for CRP
K_{CRPr}	Half maximum conc. for CRP
$V_{Co-icdabeg}$	Cooperatively coefficient for Co
$K_{Co-icdabeg}$	Half maximum conc. for Co
b	Basal trascription rate
V_b	Volume of bacterium
V	Volume of solution
k_g	Growth rate
x_{0max}	Maximum population

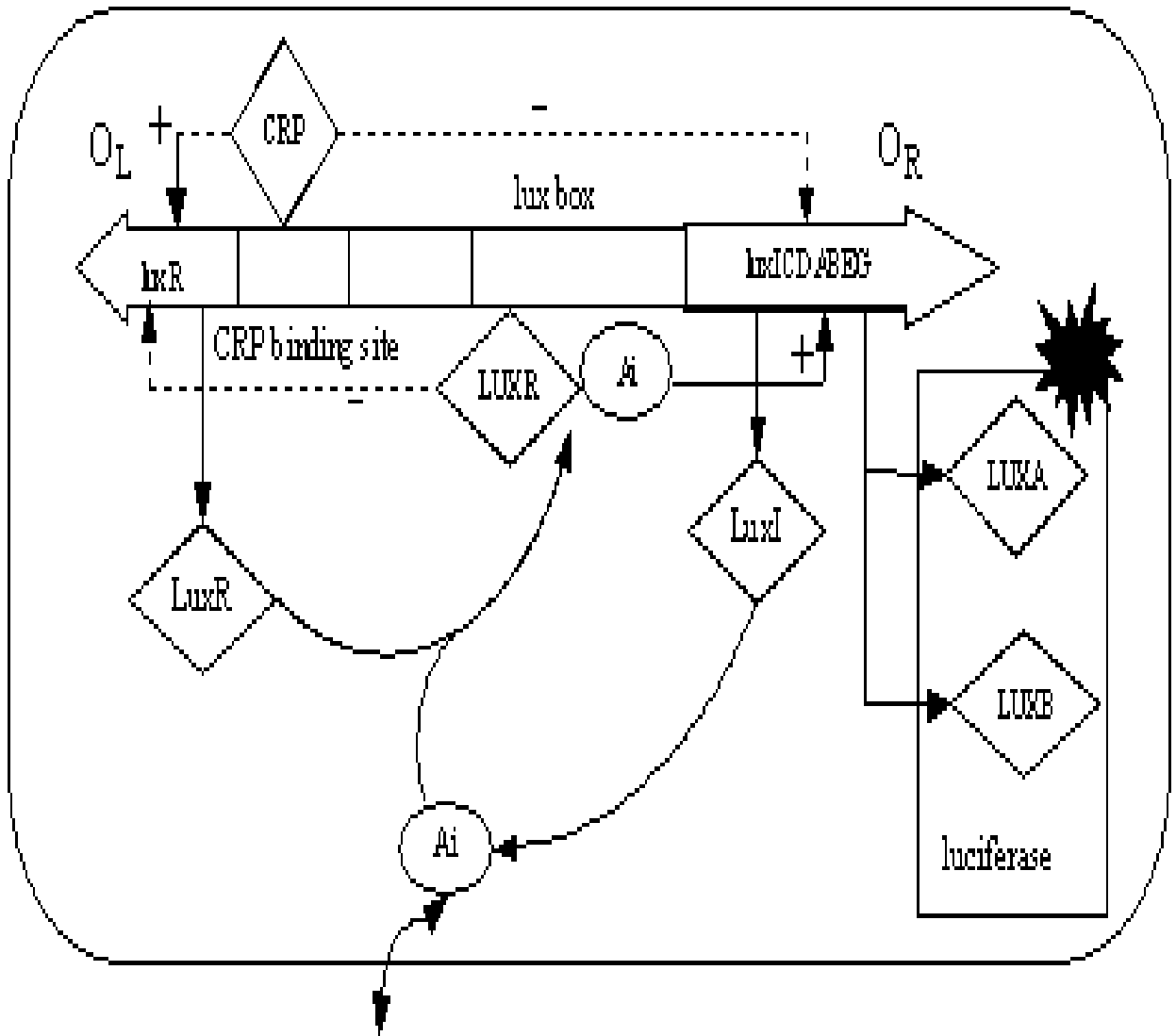


Figure 3.4. This diagram—based on a figure presented in [1:8]—depicts the regulatory control of luminescence in *V. fischeri*. Reactions take place as follows. (1) The autoinducer A_i reversibly binds to protein $LuxR$, forming a complex Co . This complex subsequently binds to the lux box. CRP (cAMP receptor protein) is assumed to reside within the cell at a constant concentration and also binds to the lux box. (2) The transcription (production of mRNA x_1) from the $luxR$ promoter (a regulatory site) is positively modulated by the binding of CRP to its binding site. The transcription of the $luxICDABEG$ (production of mRNA x_2) is positively modulated by the binding of Co to the lux box. (3) Growth in levels in CRP and Co provide negative feedback (inhibit) $luxICDABEG$ and $luxR$ transcription, respectively.

3.7.3 Budding Yeast Cell Cycle. As stated in section 2.1, a typical division cycle of a eukaryotic cell involves two main phases—S and M—temporally separated by G1 and G2 phases. Sometimes, these phases are distinct (observable), and cell division is usually symmetrical. These phases encapsulate the following events. DNA replication is initiated, eventually forming sister chromatids (S phase). These sister chromatids are then aligned onto the metaphase plate and subsequently segregated into two nuclei (M phase). Finally, cell division occurs, resulting in one mother cell and one new daughter cell. The mother cell now resides in G1 phase, pending the initiation of another division cycle.

Some characteristics of the cell cycle of budding yeast differ from the behavioral events described previously. Budding yeast cells divide asymmetrically (versus symmetrically). Consequently, the mass of the daughter cell is proportionally smaller than the mother cell. Another dissimilarity is that budding yeast cells “progress simultaneously through S and M phases (DNA synthesis, spindle formation, and chromosome alignment), without any noticeable condensation of chromosomes [1:370].” Nonetheless, the division cycle for eukaryotes is closely regulated to ensure that the correct order and strict alternation of S and M phases are enforced (e.g., DNA replication is complete before separation of sister chromatids is initiated.).

“Major cell cycle events [DNA synthesis, bud emergence, spindle formation, nuclear division, and cell separation] in budding yeast are controlled by a single CDK [cyclin-dependent kinase] (Cdc28) in conjunction with two families of cyclins: Cln1-3 and Clb1-6 [1:370]”. CDK binds with one of the cyclin members to form a heterodimer (e.g., Clb5/Cdc28). This binding ‘activates’ the complex and enables it to carry out its regulatory activities. Chen et al. list the functions of each heterodimer [8:370].

- Cln1/Cdc28 and Cln2/Cdc28 influence budding and spindle pole body duplication.

- Cln3/Cdc28 influences the size at which newborn cells execute Start.
- Clb5/Cdc28 and Clb6/Cdc28 influence timely DNA replication.
- Clb3/Cdc28 and Clb4/Cdc28 influence DNA replication and spindle formation.
- Clb1/Cdc28 and Clb2/Cdc28 influence proper completion of mitosis.

Novak partitions the above heterodimers into two different sets. CycB/Cdc28 is the set that contains a *Clb* part [20:2]. Cln/Cdc28 is the set of that contains a *Cln* part [20:4].

Cell division events are regulated by the interaction of CycB/Cdc28 with what Novak terms as enemies and helper molecules. There are two types of helper molecules—helpers of the enemies and helpers of CycB/Cdc28. The enemies and their helper molecules collectively inhibit the activity of CycB/Cdc28. On the other hand, helper molecules can reinitiate Cdk/CycB activity (when a certain condition is satisfied), and both eventually inactivate enemies and their friends. Table 3.3 summarizes various molecular interactions that modulate CycB/Cdc28 activity [20:2–4].

Table 3.3. Cdk/CycB Molecular Interactions

<i>Name</i>	<i>Relation Type</i>	<i>Interaction Effect</i>
Cdh1	Enemy	Promotes Cdk/CycB degradation
Sic1	Enemy	Binds to Cdk/CycB, causing inactivation
Cdc20	Enemy helper	Promotes Cdk/CycB degradation
Cdk/Cln kinases	Cdk/CycB helper	Assists in initiating Start transition

Due to the antagonistic interaction between CycB/Cdc28 and enemies, the control system for cell division operates from two alternative states—*State*₁ and *State*₂. CDK activity (combined activities of Clb’s and Cln’s) is low in *State*₁, while their enemies—Sic1 and Cdh1 are high. Therefore, concentration levels of the Clb’s and Cln’s remain mostly low during *State*₁. On the other hand, CDK activity is high in *State*₂. As a result, the concentrations of the Clb’s and Cln’s are high, where their enemies’ concentrations remain mostly low during *State*₂. Along these lines, Chen et al. cite the following quote [8:370].

Nasmyth (1996) has proposed that the heart of the budding yeast cell cycle is an alternation between two self-maintaining states: the G1 state, in which APC is active, CDK activity is low, and origins are licensed; and the S/M state, in which APC is shut off, CDK activity is high, and origins are fired and incapable of firing again.

Control is transferred between self-maintaining states by means of enabled transitions. The budding yeast cell cycle has two major transitions—**Start** and **Finish**. **Start** is enabled when the cell reaches critical mass. Once **Start** is enabled, Cln/Cdk helper molecules initiate the buildup of CycB/Cdc28 molecules by activating transcription factors that synthesize CycB molecules (the concentration of Cdc28 is assumed to exist in excess). The combined concentration of CycB/Cdc28 and Cln/Cdc28 molecules eventually increase to a level that allows them to inhibit the activity of the enemies and their helper molecules. At this stage, the environment is primed for a new round of DNA replication.

Finish is enabled when DNA is fully replicated *and* chromosomes are aligned on the metaphase plate. Once **Finish** is enabled, Cdc20—helper molecule for the enemies—is activated. Indirectly Cdc20 promotes dissociation of sister chromatids, activation of Hct1 (partly responsible for the degradation of Clb2/Cdc28), and activation of Swi5 (a transcription factor that initiates the synthesis of Sic1) [8:373]. Ultimately, increasing activity levels of the enemies and their helper molecules nullify CycB/Cdc28 activity. Figure yeastcycle is the molecular model for control of CDK activities. It is extracted from the consensus model published by Chen et al. [8:373]. Chen et al. list the following events that occur when either transition is initiated. Detailed information on the cell division cycle for budding yeast can be found in [8:369–391].

Due to the advanced nature of deriving the computational model for characterizing system control for the cell division cycle of budding yeast, I omit steps 1-3 as described in section 3.2. The complete set of reaction equations is listed in appendix A. Chen et al. present the following system of ODEs that characterizes cell division

control for budding yeast [8:374].

$$\begin{aligned}
\frac{d}{dt}[Cln2] &= (k'_{s,n2} + k''_{s,n2}[SBF])mass - k_{d,n2}[Cln2] \\
\frac{d}{dt}[Clb2]_T &= (k'_{s,b2} + k''_{s,b2}[Mcm1])mass - V_{d,b2}[Clb2]_T \\
\frac{d}{dt}[Clb5]_T &= (k'_{s,b5} + k''_{s,b5}[MBF])mass - V_{d,b5}[Clb5]_T \\
\frac{d}{dt}[sic1]_T &= k'_{s,c1} + k''_{s,c1}[Swi5] - (k_{d1,c1} + \frac{V_{d2,c1}}{J_{d2,c1}+[sic1]_T}[sic1]_T) \\
\frac{d}{dt}[Clb2/Sic1] &= k_{as,b2}[Clb2][sic1] - \left(k_{di,b2} + V_{d,b2} + K_{d1,c1} + \frac{V_{d2,c1}}{J_{d2,c1}+[sic1]_T}\right) [Clb2/Sic1] \\
\frac{d}{dt}[Clb5/Sic1] &= k_{as,b2}[Clb5][sic1] - \left(k_{di,b5} + V_{d,b5} + K_{d1,c1} + \frac{V_{d2,c1}}{J_{d2,c1}+[sic1]_T}\right) [Clb5/Sic1] \\
\frac{d}{dt}[Cdc20]_T &= (k'_{s,20} + k''_{s,20}[Clb2]) - k_{d,20}[Cdc20]_T \\
\frac{d}{dt}[Cdc20] &= k_{a,20}([Cdc20]_T - [Cdc20]) - (V_{i,20} + k_{d,20})[Cdc20] \\
\frac{d}{dt}[Hct1] &= \frac{(k'_{a,t1} + k''_{a,t1}[Cdc20])([Hct1]_T - [Hct1])}{J_{a,t1} + [Hct1]_T - [Hct1]} - \frac{V_{i,t1}[Hct1]}{J_{i,t1} + [Hct1]} \\
\frac{d}{dt}[mass] &= \mu(mass) \\
\frac{d}{dt}[ORI] &= k_{s,ori}([Clb5] + \varepsilon_{ori,b2}[Clb2]) - k_{d,ori}[ORI] \\
\frac{d}{dt}[BUD] &= k_{s,bud}([Cln2] + [Cln3]^* + \varepsilon_{bud,b5}[Clb5]) - k_{d,bud}[BUD]
\end{aligned} \tag{3.66}$$

where

$$\begin{aligned}
[Clb2]_T &= [Clb2] + [Clb2/Sic1] \\
[Clb5]_T &= [Clb5] + [Clb5/Sic1] \\
[Sic1]_T &= [sic1] + [Clb2/Sic1] + [Clb5/Sic1]
\end{aligned} \tag{3.67}$$

3.7.4 Glycolysis. “Glycolysis is the metabolic pathway used by most autotrophic and heterotrophic organisms, both aerobes and anaerobes, to begin to break down glucose [6:114].” In the Embden-Meyerhof pathway, the breakdown of glucose occurs in 10 steps or catalyzed reactions [6:A27]. These 10 steps encompass three stages:

1. Confine substrate (glucose) inside the cell and form phosphorylated six-carbon units (fructose 1, 6-diphosphate).
2. A six-carbon unit is split into two three-carbon units, and two ATP molecules are produced. Two NAD molecules are reduced.
3. Pyruvic acid is formed, and two more ATP molecules are produced. [6:A27]

Figure A illustrates this metabolic pathway.

Each reaction (step) in a metabolic pathway is controlled by a particular enzyme. Ten enzymes in the Embden-Meyerhof pathway are the terms in figure A that contain the suffix 'ase'. As previously stated in section 2.3.4, a single catalyzed reaction can be divided into two separate reactions. In that section, equations 2.10 and 2.11 are derived from equation 2.9. Equations 2.13 and 2.14 are derived from equation 2.12. Along these lines, the set of 10 catalyzed reactions is represented by a set of 20 equivalent reaction equations. Table 3.5 lists the set of 10 reaction equations derived from figure A. Table 3.4 indicates the correspondence between the entries depicted in figure A and the y_i 's (for $i = 1$ to 37) which appear in tables 3.5, 3.6, 3.7, and equation 3.68. Table 3.6 lists the resultant set of 20 reaction equations that is derived from table 3.5. Each pair of reaction equations in table 3.6 corresponds to a single reaction in table 3.5. For example, reactions 1A and 1B in table 3.6 correspond to reaction 1 in table 3.5, and reactions 2A and 2B correspond to reaction 2, and so on.

Each $CatTran_j$ entry (for $j = 1$ to 10) in table 3.4 is an enzyme-substrate complex that corresponds to a reaction in table 3.5. For each one-way reaction in table 3.5, each $CatTran_j$ is analogous to ES in equations 2.10 and 2.11. For each two-way reaction in table 3.5, each $CatTran_j$ is analogous to ES in equations 2.13 and 2.14. The v_k 's (for $k = 1$ to 20), which appear in equation 3.68, are defined in table 3.7.

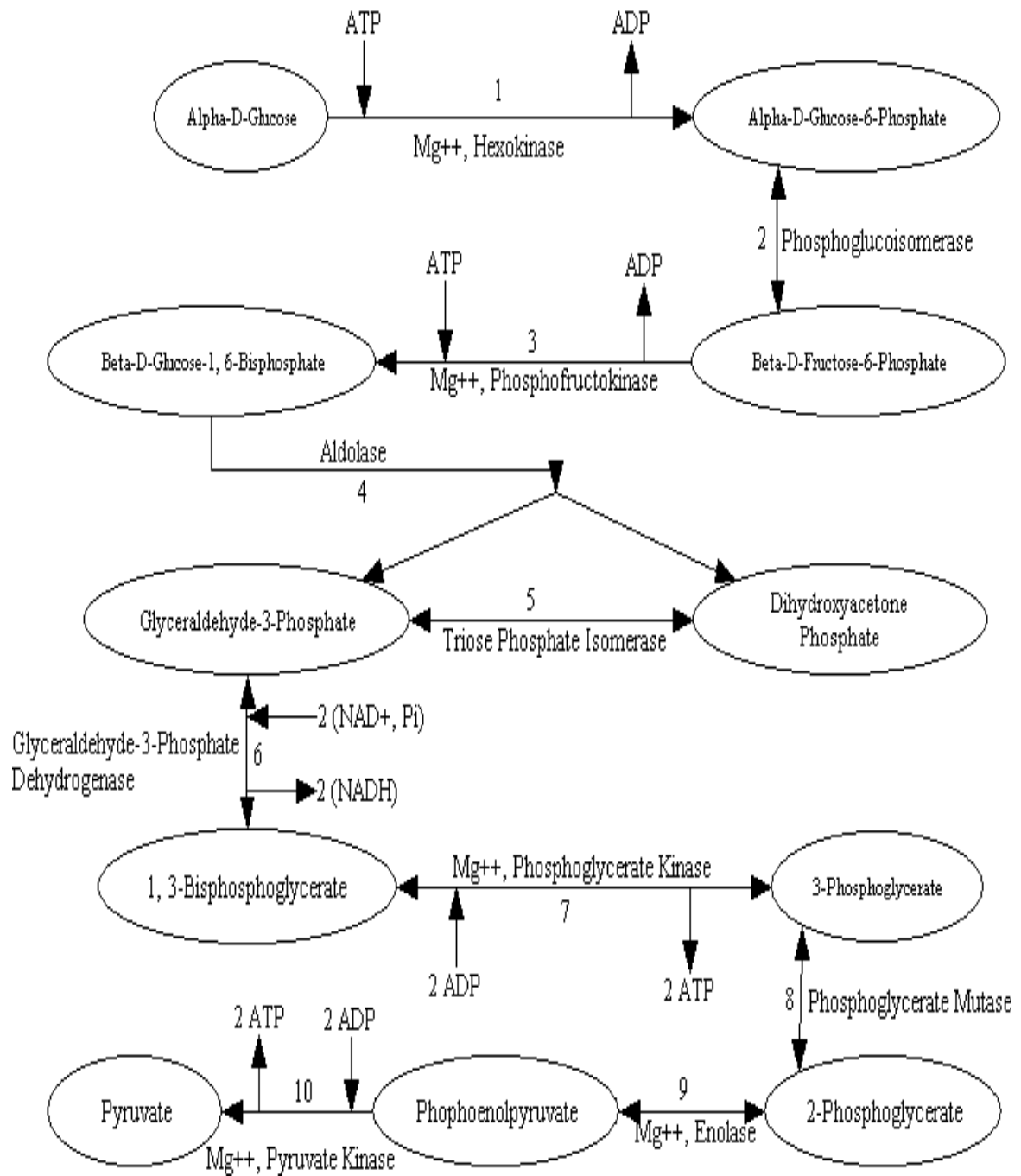


Figure 3.6. The glycolytic pathway involves 10 catalyzed reactions. As a result glucose is broken down to pyruvate [6:A27].

I derive the following system of ODEs from tables 3.4, 3.6, and 3.7.

$$\frac{d[y_1]}{dt} = v_1$$

$$\frac{d[y_2]}{dt} = v_1 + v_5 - v_{14} - v_{20}$$

$$\frac{d[y_3]}{dt} = -v_2 - v_6 + v_{13} + v_{19}$$

$$\frac{d[y_4]}{dt} = v_1 - v_2 + v_5 - v_6 + v_{13} - v_{14} + v_{17} - v_{18} + v_{19} - v_{20}$$

$$\frac{d[y_5]}{dt} = v_1 - v_2$$

$$\frac{d[y_6]}{dt} = -v_1 + v_2$$

$$\frac{d[y_7]}{dt} = -v_2 + v_3$$

$$\frac{d[y_8]}{dt} = v_3 - v_4$$

$$\frac{d[y_9]}{dt} = -v_3 + v_4$$

$$\frac{d[y_{10}]}{dt} = -v_4 + v_5$$

$$\frac{d[y_{11}]}{dt} = v_5 - v_6$$

$$\frac{d[y_{12}]}{dt} = -v_5 + v_6$$

$$\frac{d[y_{13}]}{dt} = -v_6 + v_7$$

$$\frac{d[y_{14}]}{dt} = v_7 - v_8$$

$$\frac{d[y_{15}]}{dt} = -v_7 + v_8$$

$$\frac{d[y_{16}]}{dt} = -v_8 - v_{10} + v_{11}$$

$$\frac{d[y_{17}]}{dt} = -v_8 + v_9$$

$$\frac{d[y_{18}]}{dt} = v_9 - v_{10}$$

$$\frac{d[y_{19}]}{dt} = -v_9 + v_{10}$$

$$\frac{d[y_{20}]}{dt} = v_{11}$$

$$\frac{d[y_{21}]}{dt} = v_{11}$$

$$\frac{d[y_{22}]}{dt} = -v_{12}$$

$$\begin{aligned}
\frac{d[y_{23}]}{dt} &= v_{11} - v_{12} \\
\frac{d[y_{24}]}{dt} &= -v_{11} + v_{12} \\
\frac{d[y_{25}]}{dt} &= -v_{12} + v_{13} \\
\frac{d[y_{26}]}{dt} &= v_{13} - v_{14} \\
\frac{d[y_{27}]}{dt} &= -v_{13} + v_{14} \\
\frac{d[y_{28}]}{dt} &= -v_{14} + v_{15} \\
\frac{d[y_{29}]}{dt} &= v_{15} - v_{16} \\
\frac{d[y_{30}]}{dt} &= -v_{15} + v_{16} \\
\frac{d[y_{31}]}{dt} &= -v_{16} + v_{17} \\
\frac{d[y_{32}]}{dt} &= v_{17} - v_{18} \\
\frac{d[y_{33}]}{dt} &= -v_{17} + v_{18} \\
\frac{d[y_{34}]}{dt} &= -v_{18} + v_{19} \\
\frac{d[y_{35}]}{dt} &= v_{19} - v_{20} \\
\frac{d[y_{36}]}{dt} &= -v_{19} + v_{20} \\
\frac{d[y_{37}]}{dt} &= -v_{20}
\end{aligned} \tag{3.68}$$

Table 3.4. Species Cross Reference List (Glycolytic Pathway)

Variable	Species
y_1	Alpha-D-Glucose
y_2	ATP
y_3	ADP
y_4	Mg ⁺⁺
y_5	Hexokinase
y_6	<i>CatTran</i> ₁
y_7	Alpha-D-Glucose-6-phosphate
y_8	Phosphoglucoisomerase
y_9	<i>CatTran</i> ₂
y_{10}	Beta-D-Fructose-6-phosphate
y_{11}	Phosphofructokinase
y_{12}	<i>CatTran</i> ₃
y_{13}	Beta-D-Fructose-1, 6-bisphosphate
y_{14}	Aldolase
y_{15}	<i>CatTran</i> ₄
y_{16}	Glyceraldehyde-3-phosphate
y_{17}	Dihydroxyacetone phosphate
y_{18}	Triose phosphate isomerase
y_{19}	<i>CatTran</i> ₅
y_{20}	NAD ⁺
y_{21}	P_i
y_{22}	NADH
y_{23}	Glyceraldehyde-3-phosphate dehydrogenase
y_{24}	<i>CatTran</i> ₆
y_{25}	1, 3-Bisphosphoglycerate
y_{26}	Phosphoglycerate kinase
y_{27}	<i>CatTran</i> ₇
y_{28}	3-Phosphoglycerate
y_{29}	Phosphoglycerate mutase
y_{30}	<i>CatTran</i> ₈
y_{31}	2-Phosphoglycerate
y_{32}	Enolase
y_{33}	<i>CatTran</i> ₉
y_{34}	Phosphoenolpyruvate
y_{35}	Pyruvate kinase
y_{36}	<i>CatTran</i> ₁₀
y_{37}	Pyruvate

Table 3.5. Reaction Equations (Glycolytic Pathway)

Reaction	Reaction Equation
1	$y_1 + y_2 + y_4 + y_5 \rightarrow y_3 + y_4 + y_5 + y_7$
2	$y_7 + y_8 \rightleftharpoons y_8 + y_{10}$
3	$y_2 + y_4 + y_{10} + y_{11} \rightarrow y_3 + y_4 + y_{11} + y_{13}$
4	$y_{13} + y_{14} \rightleftharpoons y_{14} + y_{16} + y_{17}$
5	$y_{17} + y_{18} \rightleftharpoons y_{16} + y_{18}$
6	$y_{16} + y_{20} + y_{21} + y_{23} \rightleftharpoons y_{22} + y_{23} + y_{25}$
7	$y_3 + y_4 + y_{25} + y_{26} \rightleftharpoons y_2 + y_4 + y_{26} + y_{28}$
8	$y_{28} + y_{29} \rightleftharpoons y_{29} + y_{31} + y_{17}$
9	$y_4 + y_{31} + y_{32} + y_{23} \rightleftharpoons y_4 + y_{32} + y_{34}$
10	$y_3 + y_4 + y_{34} + y_{35} \rightarrow y_2 + y_4 + y_{35} + y_{37}$

Table 3.6. Reaction Equations with Catalytic Transitions (Glycolytic Pathway)

Reaction	Reaction Equation
1A	$y_1 + y_2 + y_4 + y_5 \xrightleftharpoons[k_2]{k_1} y_6$
1B	$y_6 \xrightarrow{k_3} y_3 + y_4 + y_5 + y_7$
2A	$y_7 + y_8 \xrightleftharpoons[k_5]{k_4} y_9$
2B	$y_9 \xrightleftharpoons[k_7]{k_6} y_8 + y_{10}$
3A	$y_2 + y_4 + y_{10} + y_{11} \xrightleftharpoons[k_9]{k_8} y_{12}$
3B	$y_{12} \xrightarrow{k_{10}} y_3 + y_4 + y_{11} + y_{13}$
4A	$y_{13} + y_{14} \xrightleftharpoons[k_{13}]{k_{12}} y_{15}$
4B	$y_{15} \xrightleftharpoons[k_{15}]{k_{14}} y_{14} + y_{16} + y_{17}$
5A	$y_{17} + y_{18} \xrightleftharpoons[k_{17}]{k_{16}} y_{19}$
5B	$y_{19} \xrightleftharpoons[k_{19}]{k_{18}} y_{16} + y_{18}$
6A	$y_{16} + y_{20} + y_{21} + y_{23} \xrightleftharpoons[k_{21}]{k_{20}} y_{24}$
6B	$y_{24} \xrightleftharpoons[k_{23}]{k_{22}} y_{22} + y_{23} + y_{25}$
7A	$y_3 + y_4 + y_{25} + y_{26} \xrightleftharpoons[k_{25}]{k_{24}} y_{27}$
7B	$y_{27} \xrightleftharpoons[k_{27}]{k_{26}} y_2 + y_4 + y_{26} + y_{28}$
8A	$y_{28} + y_{29} \xrightleftharpoons[k_{29}]{k_{28}} y_{30}$
8B	$y_{30} \xrightleftharpoons[k_{31}]{k_{30}} y_{29} + y_{31}$
9A	$y_4 + y_{31} + y_{32} \xrightleftharpoons[k_{33}]{k_{32}} y_{33}$
9B	$y_{33} \xrightleftharpoons[k_{35}]{k_{34}} y_4 + y_{32} + y_{34}$
10A	$y_3 + y_4 + y_{34} + y_{35} \xrightleftharpoons[k_{37}]{k_{36}} y_{36}$
10B	$y_{36} \xrightarrow{k_{38}} y_2 + y_4 + y_{35} + y_{37}$

Table 3.7. Derived Reaction Velocities for Embden-Meyerhof Glycolytic Pathway

Reaction	Velocity
1A	$v_1 = -k_1 y_1 y_2 y_4 y_5 + k_2 y_6$
1B	$v_2 = -k_3 y_6$
2A	$v_3 = -k_4 y_7 y_8 + k_5 y_9$
2B	$v_4 = -k_6 y_9 + k_7 y_8 y_{10}$
3A	$v_5 = -k_8 y_2 y_4 y_{10} y_{11} + k_9 y_{12}$
3B	$v_6 = -k_{10} y_{12}$
4A	$v_7 = -k_{11} y_{13} y_{14} + k_{12} y_{15}$
4B	$v_8 = -k_{13} y_{15} + k_{14} y_{14} y_{16} y_{17}$
5A	$v_9 = -k_{15} y_{17} y_{18} + k_{16} y_{19}$
5B	$v_{10} = -k_{17} y_{19} + k_{18} y_{16} y_{18}$
6A	$v_{11} = -k_{19} y_{16} y_{20} y_{21} y_{23} + k_{20} y_{24}$
6B	$v_{12} = -k_{21} y_{24} + k_{22} y_{22} y_{23} y_{25}$
7A	$v_{13} = -k_{23} y_3 y_4 y_{25} y_{26} + k_{24} y_{27}$
7B	$v_{14} = -k_{25} y_{27} + k_{26} y_2 y_4 y_{26} y_{28}$
8A	$v_{15} = -k_{27} y_{28} y_{29} + k_{28} y_{30}$
8B	$v_{16} = -k_{29} y_{30} + k_{30} y_{29} y_{31}$
9A	$v_{17} = -k_{31} y_4 y_{31} y_{32} + k_{32} y_{33}$
9B	$v_{18} = -k_{33} y_{33} + k_{34} y_4 y_{32} y_{34}$
10A	$v_{19} = -k_{35} y_3 y_4 y_{34} y_{35} + k_{36} y_{36}$
10B	$v_{20} = -k_{37} y_{36}$

IV. Computational Model Results

4.1 Overview

In chapter III, I discuss the following reaction models: modified Brusselator, lysis-lysogeny pathway of a mutant bacteriophage, quorum sensing system that controls luminescence in *V. fischeri*, control system for cyclin dependent kinases (CDKs) in the budding yeast cell cycle, and a glycolytic pathway. One way of shedding light on the overall functionality of these reaction systems is to derive a computational model (system of ODEs) for each reaction system. Computational models are generally used to track how state variables (concentrations of each species) evolve over a specified time interval. The rate-equation approach for deriving computational models is the method exclusively used in this thesis. This approach involves the following steps.

1. Convert a wiring diagram into a set of reaction equations (e.g., table 3.6).
2. Derive individual reaction rate velocities from the set of reaction equations derived in step 1 (e.g., table 3.7).
3. Express each state variable (species) with the appropriate subset of reaction rate velocities derived in step 2 (e.g., equation 3.68).

BioCharon and JigCell are software tools that serve as aids in building biomolecular models, deriving (via rate-equation approach) computational models, and simulating these models. Steps 1-3 are automated in BioCharon. Steps 2-3 are automated in JigCell.

In this chapter, I present general simulation results that highlight certain qualitative features (e.g., conservation relation among species) of the previously mentioned reaction models. I then compare simulation results from MATLAB, BioCharon, and/or JigCell. Metrics for these comparisons are explained in section 4.2.

Simulation results are obtained by MATLAB, BioCharon, and JigCell for the modified Brusselator model and lysis-lysogeny pathway. Simulation results for the quorum sensing system are obtained by MATLAB and JigCell. This system is not simulated using BioCharon, because the cost is extremely high in simulation time and memory (an extremely small integration step size is required to obtain accurate results). Simulation results for the control system of CDKs in the budding yeast cell division cycle are obtained from MATLAB and JigCell. An instance of this model is not created using BioCharon, because the user is not able to explicitly define—via Bio Sketch Pad—key aspects of this model. For example, the user is not able to explicitly define Michaelis-Menten rate laws and functions for transcription factors (e.g., $V_{d,b2}$ and $V_{d,b5}$ in equation 3.66). Simulation results for the glycolytic pathway are solely obtained by MATLAB, because the computational time for MATLAB is substantially less than that for both BioCharon and JigCell.

4.2 *Experimental Design*

All or some of the following metrics are used in comparing simulation results: absolute error, sum of absolute errors, and relative error. *Absolute error* is the absolute value of the difference between simulation results obtained from MATLAB and either BioCharon or JigCell. This metric is used in all comparisons.

Sum of absolute errors is the sum of individual absolute errors among all state variables (species) for a specified trial. This metric is used during the comparison of simulation results for the lysis-lysogeny pathway in order to ascertain overall accuracy for this reaction system. This metric is used solely in table 4.9.

Relative error is used in comparing simulation results for the quorum sensing system. This additional metric is used because simulation results among species vary several orders of magnitude. Absolute error among species having significantly larger values is more likely to have larger absolute error than among species with significantly smaller values. For example, luxICDABEG has a value of 9.80402, but

LuxR has a value of 5208.60414 in table 4.12. Relative error is used to better gauge the degree of error among all species. Relative error is $\frac{|MATLAB_{Result} - Other_{Result}|}{|MATLAB_{Result}|}$. This metric is used in tables 4.11, 4.12, 4.13, 4.14, 4.15, and 4.16.

“With a user community more than 500,000 strong spread throughout industry, government, and academia, MATLAB is the recognized standard worldwide for technical computing [16:2]”. In this regard, simulation results obtained from MATLAB are regarded as baseline results (i.e., error = 0). Therefore, the accuracy of simulation results generated from BioCharon and JigCell are gauged from simulation results generated by MATLAB.

System parameters for BioCharon, JigCell, and MATLAB are configured in the following manner. The integration step size for the ordinary differential equation (ODE) solver of BioCharon is set to 0.01 during all simulations. I performed several pilot studies in order to decide which step size to use. A smaller step size may slightly increase accuracy (depending on the system); however, total simulation time is greatly increased as the integration step size is shortened. The ODE solver for JigCell is set to ‘stiff’ during all simulations (a solver that is specifically designed to approximate solutions for a stiff system of differential equations). The ODE solver for MATLAB is set ‘ode23s’ at the default accuracy (relative tolerance of 1E-3 and vector of absolute error tolerances of 1E-6 by default). The system parameter that adjusts total simulation time is synchronized for all software tools.

In obtaining simulation results, each of the five computational models is submitted as a workload to the appropriate software tool. Each workload submission on either MATLAB, BioCharon, and JigCell has an accompanying set of initial conditions (species concentrations and model parameters). Initial concentrations for species are varied in sections 4.3, 4.4, 4.5, and 4.6. While maintaining the same initial concentrations for species, a single model parameter is varied in section 4.7.

4.3 Modified Brusselator Model

The Brusselator model “reveals simple oscillations and damped oscillations by the suitable parameters [18:249].” I describe a modified version of the Brusselator model in section 3.6. The modification involves the addition of a free parameter ϵ . This parameter reveals the *strength* of the autocatalytic process for species X (equation 3.11) [18:249]. If ϵ is assigned an integer value greater than or equal to one and model parameters a and b are defined using equations 3.35 and 3.28, respectively, the critical point (\hat{x}, \hat{y}) —defined by equation 3.26—is stable.

Figure 4.1 illustrates 10 critical points for the modified Brusselator system. Each critical point is obtained by using equation 3.26 and imposing the above constraints for parameters a , b and ϵ . The integer value of ϵ is varied from 1 to 10. It is clear from figure 4.1 that the critical point concentration for species x increases as ϵ is increased. This behavior is consistent, because ϵ serves as the autocatalytic power control for x . It follows that as the level of ϵ is increased, the operating level of x is increased. Figure 4.1 also illustrates that the relative effect of ϵ on x diminishes as ϵ is increased.

4.3.1 General Results. Figures 4.2 and 4.3 illustrate numerical simulations for the Brusselator system performed on MATLAB. The value for parameters a and b are selected in accordance with the previously mentioned constraints that ensure a stable critical point. In both cases, $a = 0.7$, $b = 0.5$, and $\epsilon = 1$. In figure 4.2, the initial concentrations of x and y are zero. In figure 4.3 the initial concentrations for both species are five. Despite the significant difference between initial conditions, the resulting trajectory is back to the same critical point: $\hat{x} = 0.7$ and $\hat{y} \approx 1.4$. This behavior of both trajectories returning to the same critical point is illustrated in figure 4.4 (phase plane plots). The behavior of returning back to a critical point after applying large perturbations away from that critical point alludes to the possible existence of a global critical point for the modified Brusselator reaction system.

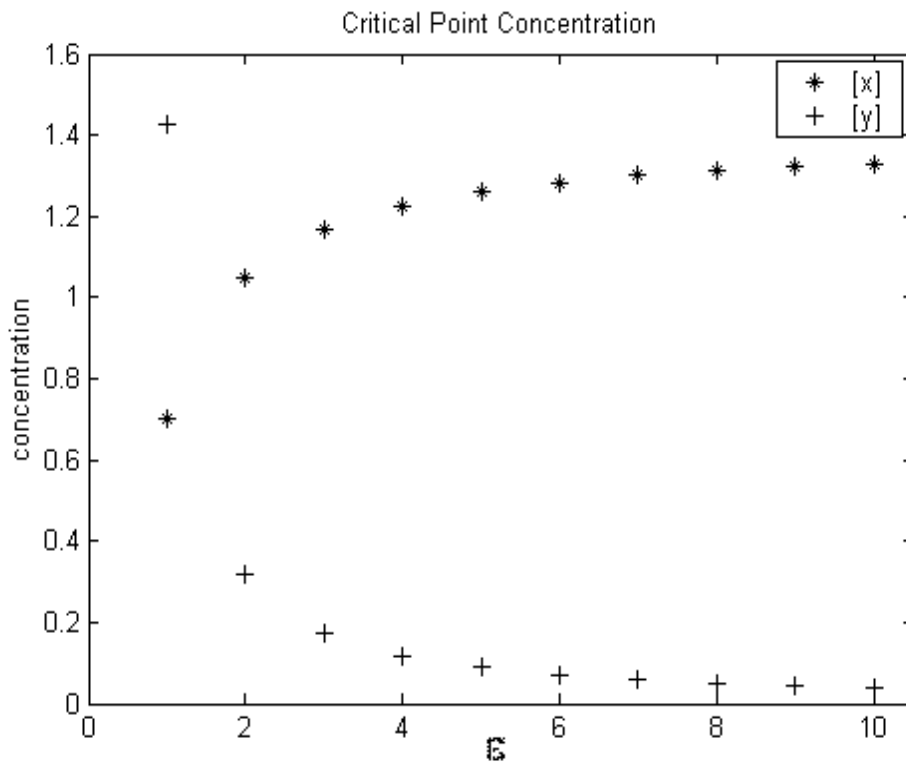


Figure 4.1. As ϵ is increased (with $a = 0.7$), the critical point solution for species x increases and y decreases. It appears that as ϵ approaches 10, the critical point solution approaches $x = 1.33$ and $y = 0.04$.

4.3.2 Comparison Results. A set of three simulations are performed using MATLAB, BioCharon, and JigCell. The parameters are a , b , x , and y , and the single factor—a parameter that is varied in each trial—is ϵ . The values of a and b are selected to ensure a stable critical point. For all three trials, the concentration of a is held constant at 0.7, and the initial concentrations of x and y are zero. The constant concentration of b is dependent on the single factor ϵ (via equation 3.28). This factor is assigned integer values 1 through 3. These parameter and factor values are listed in table 4.1.

Table 4.2 lists simulation results from MATLAB, BioCharon, and JigCell for species x and y . These results are critical point approximations (i.e., the rate of change for all state variables is zero) for the modified Brusselator system. The

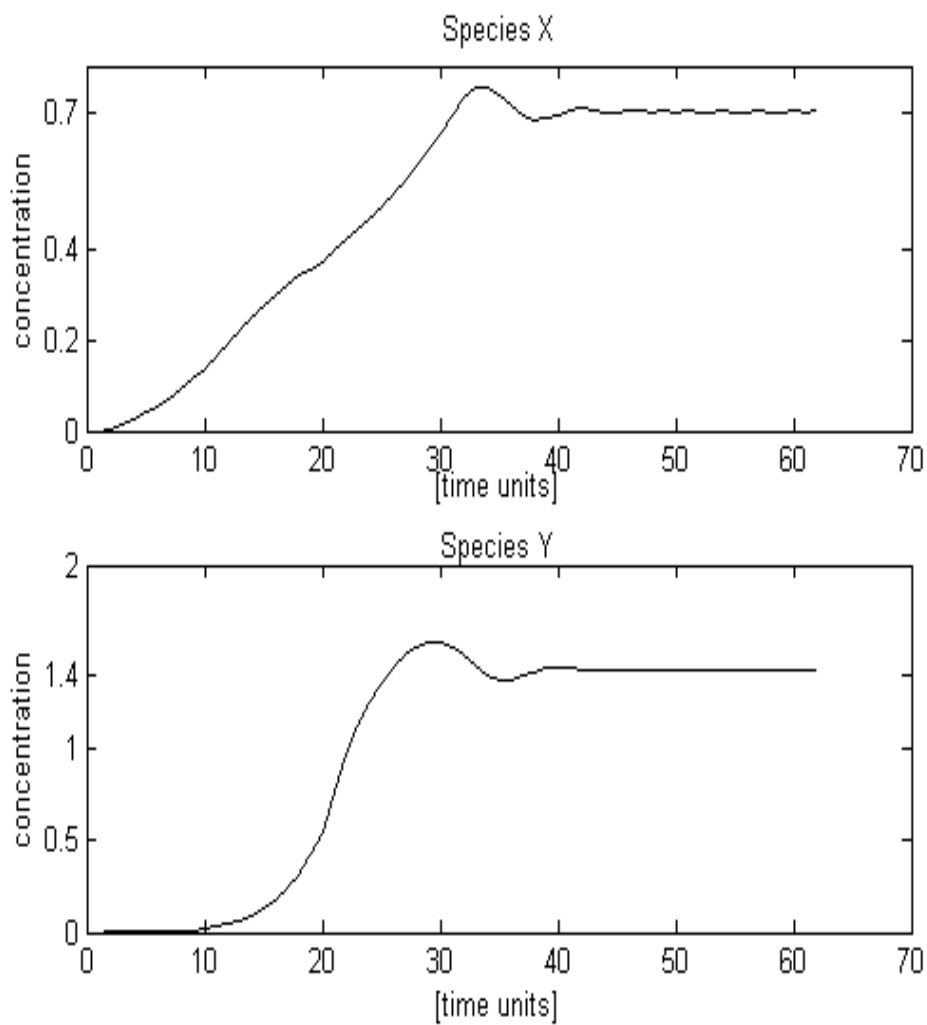


Figure 4.2. Parameter settings for the modified Brusselator model are $a = 0.7$, $b = 0.5$, and $\epsilon = 1$. Initial concentrations are $x = 0$ and $y = 0$.

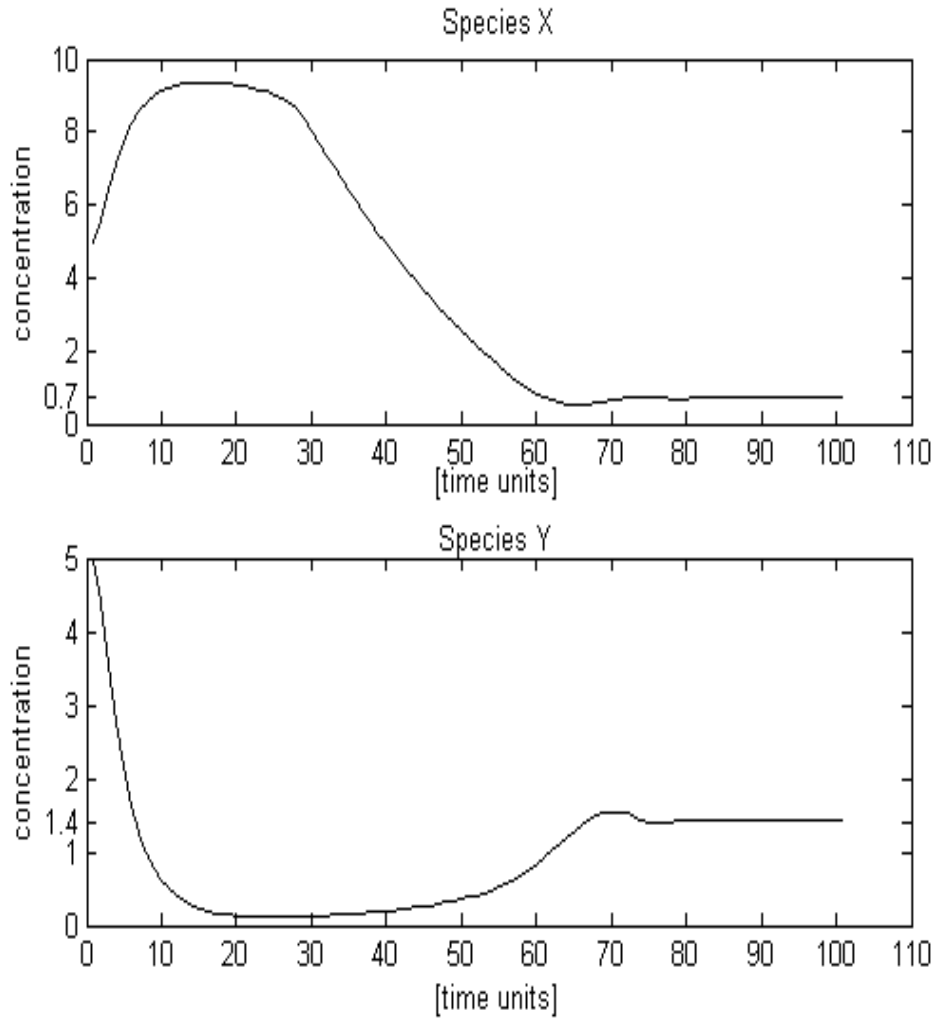


Figure 4.3. Parameter settings for the modified Brusselator model are $a = 0.7$, $b = 0.5$, and $\epsilon = 1$. Initial concentrations are $x = 5$ and $y = 5$.

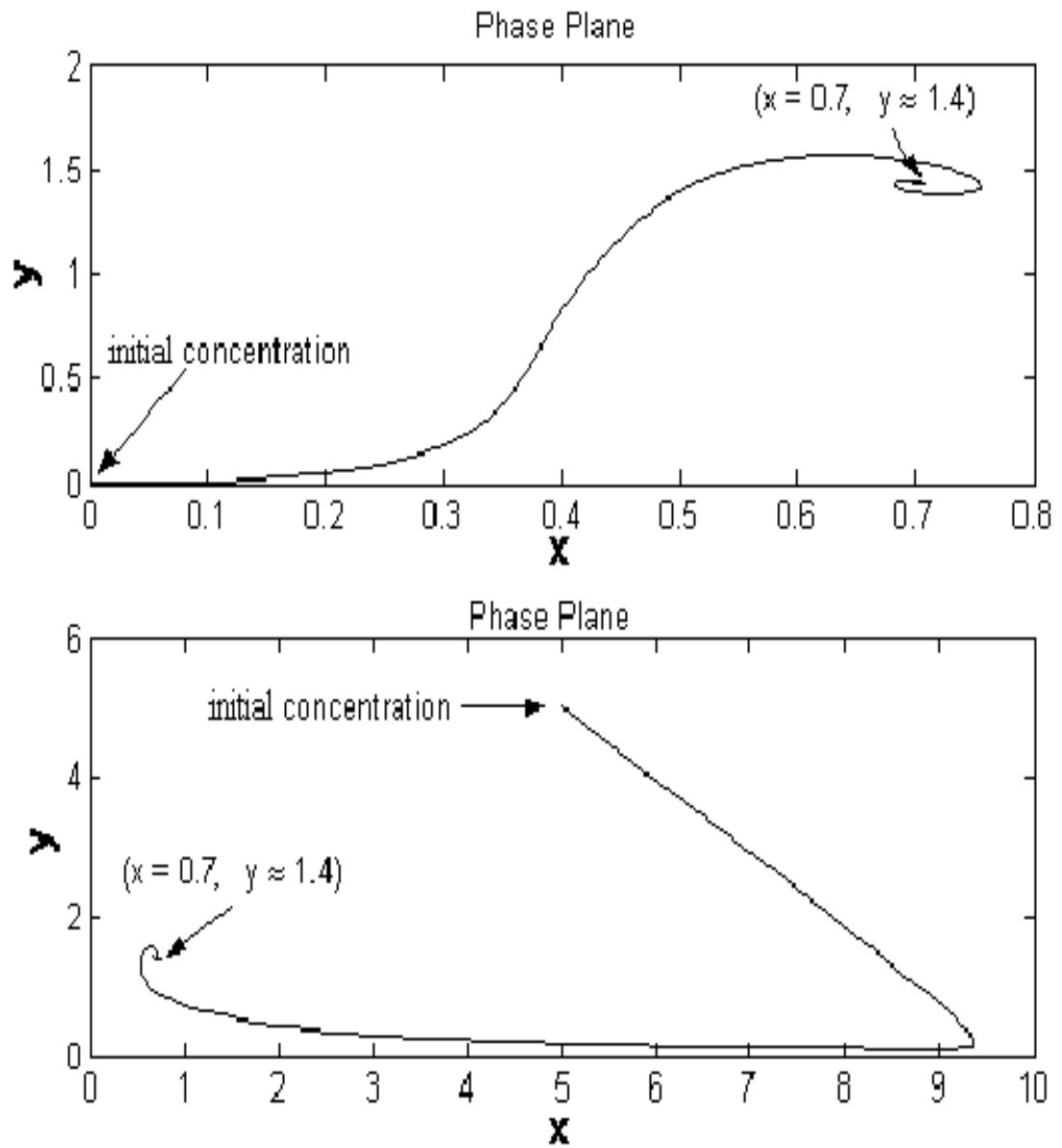


Figure 4.4. Parameter settings for the modified Brusselator model are $a = 0.7$, $b = 0.5$, and $\epsilon = 1$. The initial concentrations for the top plot are $x = 0$ and $y = 0$, and the initial concentrations for the bottom plot are $x = 5$ and $y = 5$.

results in that table are separated by trial numbers 1 through 3. Critical point approximations for each software tool are identical: $\hat{x} = 1.42857$ and $\hat{y} = 1.42857$ for trials 1 through 3. Therefore, absolute error for all critical point approximations is zero.

Table 4.1. Initial Conditions (Modified Brusselator Model)

Trial	A	B	X	Y	ϵ
1	0.7	1.0	0	0	1
2	0.7	$\frac{1}{3}$	0	0	2
3	0.7	0.2	0	0	3

Table 4.2. Critical Point Results (Modified Brusselator Model)

Trial	Application	X	Y
1	MATLAB	0.70000	1.42857
	BioCharon	0.70000	1.42857
	JigCell	0.70000	1.42857
2	MATLAB	1.05000	0.31746
	BioCharon	1.05000	0.31746
	JigCell	1.05000	0.31746
3	MATLAB	1.16667	0.17143
	BioCharon	1.16667	0.17143
	JigCell	1.16667	0.17143

4.4 Lysis-Lysogeny Pathway

Depending on which operator site the repressor (X) binds to, the mutant phage (discussed in section 3.7.1) is either virulent or temperate. When X binds to $OR2$, represented by DX_2 , the synthesis of repressor is enhanced. If X binds to $OR3$, represented by DX_2^* , the synthesis of X is inhibited. High concentration levels of X induce a temperate phage. On the other hand, low concentration levels of repressor induce a virulent phage [10:271].

4.4.1 General Results. From equation 3.61, it is obvious that the lysis-lysogeny pathway can have multiple critical points. Three real and two imaginary

critical points are calculated by substituting the appropriate values listed in table 4.3 into equation 3.61. Table 4.3 defines the following parameter values for the lysis-lysogeny pathway system: kinetic rate constants (k_1, k_t, r , etc.), conservation constant (d_T), constant for polymerization of species X (n), and constant concentration for RNA polymerase (P). Three real critical points, using equations 3.58, 3.59, and 3.61, are listed in table 4.4.

Table 4.3. Model Parameters (Lysis-Lysogeny Pathway)

Parameter	Value
k_1	1
k_{-1}	1
k_2	1
k_{-2}	1
k_3	1
k_4	1
k_{-4}	1
k_t	0.8
k_d	6
r	0.4
d_T	5
n	4
$P = p_0$	1.25

Table 4.4. Critical Points (Lysis-Lysogeny Pathway)

Species	Critical Point 1	Critical Point 2	Critical Point 3
X	0.09820	0.24592	0.88845
X_2	0.00964	0.06048	0.78934
D	4.90495	4.44597	1.56165
DX_2	0.04730	0.26889	1.23268
DX_2^*	0.04730	0.26889	1.23268
DX_2X_2	0.00046	0.01626	0.97300

Using the same approach as previously discussed in section 3.6, I perform stability analysis on the full and reduced systems of the lysis-lysogeny pathway (about each critical point listed in table 4.4). The full system is represented by equations

Table 4.5. Eigenvalues for Full Lysis-Lysogeny Pathway

	Critical Point 1	Critical Point 2	Critical Point 3
$Eigenvalue_1$	0	0	0
$Eigenvalue_2$	-0.02752	0.01455	-0.11060
$Eigenvalue_3$	-0.89098	-0.83369	-0.91172
$Eigenvalue_4$	-1.20181	-1.70225	-5.02094
$Eigenvalue_5$	-6.43729	-6.99262	-7.88717
$Eigenvalue_6$	-11.94908	-12.12940	-14.43544

Table 4.6. Eigenvalues for Reduced Lysis-Lysogeny Pathway

	Critical Point 1	Critical Point 2	Critical Point 3
$Eigenvalue_1$	-0.02752	0.01455	-0.11060
$Eigenvalue_2$	-0.89098	-0.83369	-0.91172
$Eigenvalue_3$	-1.20181	-1.70225	-5.02094
$Eigenvalue_4$	-6.43729	-6.99262	-7.88717
$Eigenvalue_5$	-11.94908	-12.12940	-14.43544

3.50 through 3.55. The reduced system is represented by equations 3.50, 3.56, 3.52, 3.57, and 3.54. The calculated eigenvalues from each stability analysis are listed in tables 4.5 and 4.6.

With the exception of the zero entries ($Eigenvalue_1$) for each critical point listed in table 4.5, the eigenvalues for tables 4.5 and 4.6 are identical. The zero eigenvalue listed for each critical point in table 4.5 is expected because there exists a conservation relation between equations 3.52 through 3.55 (i.e., the full system can be reduced). From table 4.6, the stability about (near) each critical point can be ascertained (recall: in the reduced system, $[DX_2X_2] = d_T - ([D] + [DX_2] + [DX_2^*])$).

As a result, critical points one and three are asymptotically stable. This is true because all calculated eigenvalues about critical points one and three are negative [7:151]. In contrast, critical point two is not asymptotically stable, because $Eigenvalue_1$ is positive. As a reminder, small perturbations away from either stable critical point result in a trajectory back to the respective critical point.

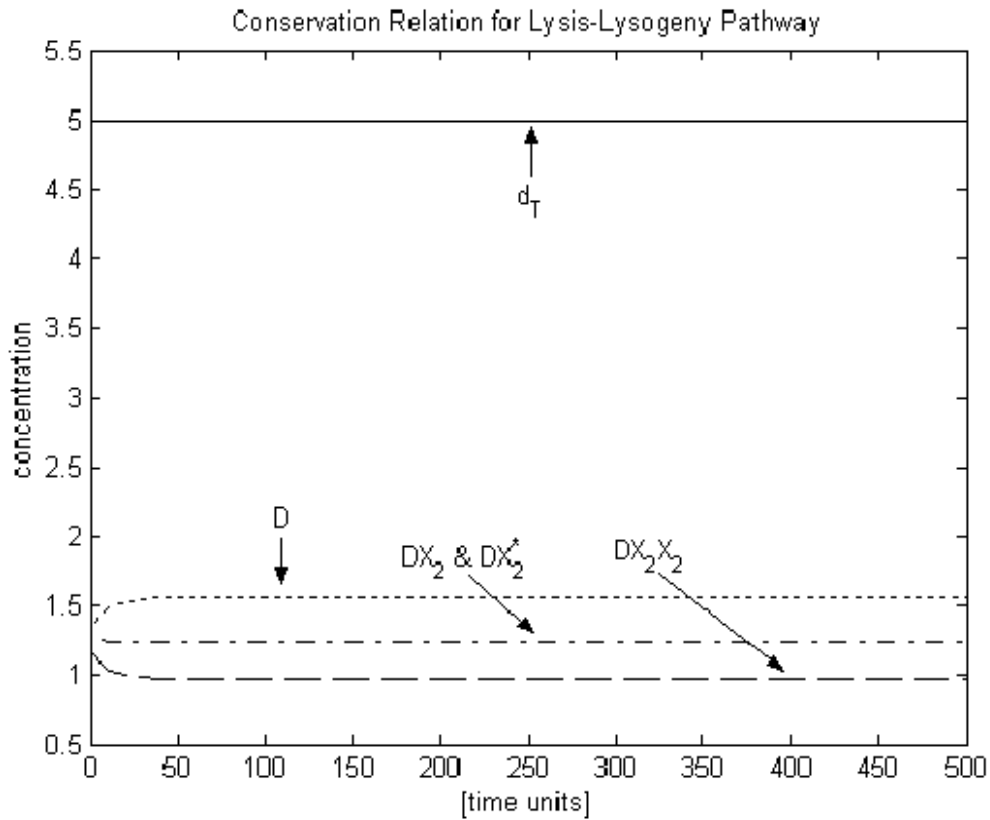


Figure 4.5. This illustrates the conservation relation: $d_T = ([D] + [DX_2] + [DX_2^*] + [DX_2X_2])$. It is apparent that the sum of concentration levels among these species remains constant as the system evolves. In this case, $d_T = 5$. Parameter settings and initial conditions are used from tables 4.3 and 4.7 (trial 4).

However, small perturbations away from the unstable critical point result in a trajectory to one of the stable critical points.

The existence of a conservation relation among equations 3.52 through 3.55 implies that even though individual concentrations of D , DX_2 , DX_2^* , and DX_2X_2 may change over time, the sum of their concentrations remain constant. Figure 4.5 illustrates this conservation rule. The plot (parameter values and initial conditions are from tables 4.3 and 4.7 (trial 4), illustrated in figure 4.5, exemplifies how the total sum of concentrations among species D , DX_2 , DX_2^* , and DX_2X_2 remain at a constant value (d_T) as the modified Brusselator system evolves.

4.4.2 *Simulation Results.* I perform a set of four simulations (trials) using MATLAB, BioCharon, and JigCell. The parameter values listed in table 4.3 are used in each trial. Initial conditions for each trial are listed in table 4.7. Initial species concentrations in trial 1 represent a small perturbation away from critical point 1. Initial species concentrations in trials 2 and 3 represent small perturbations away from critical point 2. Initial species concentrations in trial 4 represent a small perturbation away from critical point 3.

Simulation results from MATLAB, BioCharon, and JigCell are listed in table 4.8. These results are critical point approximations. Critical point approximations for trials 1 and 4 are similar to critical points 1 and 3 (listed in table 4.4). This is expected because trials 1 and 4 are small perturbations away from stable critical points.

Critical point approximations for trials 2 and 4 are similar to critical points 1 and 3, respectively (listed in table 4.4). It is not surprising that both trajectories move away from critical point 2, because both trials represent small perturbations away from an unstable critical point. In trial 2, the trajectory path is to critical point 1. In trial 3, the trajectory path is to critical point 3. Apparently, initial conditions in trial 2 and 3 result in displacements in the attraction fields of critical point 1 and 3, respectively.

Absolute error in table 4.9 is calculated from entries listed in table 4.8. Simulation results in table 4.8 are critical point approximations from all three software tools. For both tables, M = MATLAB, B = BioCharon, J = JigCell. Absolute error (E_i —defined in section 4.2) for each simulation result is listed in table 4.9 (columns 3–8). Sum of absolute error ($Total_E = E_1 + E_2 + E_3 + E_4 + E_5 + E_6$) is listed in column 9.

With the exception of trial 1, $Total_E$ for BioCharon is greater than $Total_E$ for JigCell in all trials. Maximum $Total_E$ for BioCharon and JigCell occur during trial

2. The difference of $Total_E$ between BioCharon and JigCell is 0.03750; however, $Total_E$ for BioCharon is not significant from a biological standpoint.

Table 4.7. Initial Conditions (Lysis-Lysogeny Pathway)

Trial	X	X_2	D	DX_2	DX_2^*	DX_2X_2
1	0.2000	0.1000	4.800	0.0300	0.0400	0.1300
2	0.3000	0.0700	4.2400	0.3700	0.3500	0.0400
3	0.1400	0.0100	4.4459	0.2688	0.2688	0.0165
4	0.9800	0.6800	1.4000	1.0000	1.1000	1.5000

Table 4.8. Comparison Results (Lysis-Lysogeny Pathway)

Trial	App	X	X_2	D	DX_2	DX_2^*	DX_2X_2
1	M	0.09820	0.00964	4.90495	0.04730	0.04730	0.00046
	B	0.09820	0.00964	4.90495	0.04730	0.04730	0.00046
	J	0.09820	0.00964	4.90495	0.04730	0.04730	0.00046
2	M	0.88845	0.78934	1.56165	1.23268	1.23268	0.97300
	B	0.88496	0.78316	1.56730	1.22744	1.22744	0.96128
	J	0.88845	0.78934	1.56165	1.23267	1.23267	0.97300
3	M	0.09820	0.00964	4.90494	0.04730	0.04730	0.00046
	B	0.09820	0.00964	4.90515	0.04731	0.04731	0.00046
	J	0.09820	0.00964	4.90494	0.04730	0.04730	0.00046
4	M	0.88845	0.78934	1.56164	1.23268	1.23268	0.97300
	B	0.88940	0.79104	1.56011	1.23410	1.23410	0.97622
	J	0.88845	0.78934	1.56165	1.23268	1.23268	0.97300

4.5 *V. fischeri*

4.5.1 General Results. Luminescence in *V. fischeri* (represented by the concentration of LuxA/B) is controlled by its quorum sensing system. This system has the capability to ascertain local population density of like bacteria. When the population reaches a certain density level (i.e., a quorum of bacteria), the lux genes—responsible for luminescence—are subsequently activated. In this manner, luminescence in *V. fischeri* exhibits switch-like behavior. Figure 4.6 illustrates how luminescence in *V. fischeri* is activated when a quorum is reached.

Table 4.9. Absolute Error (Lysis-Lysogeny Pathway)

Trial	App	E_1	E_2	E_3	E_4	E_5	E_6	$Total_E$
2	B	0.00349	0.00618	0.00565	0.00524	0.00524	0.01172	0.03752
	J	0.00000	0.00000	0.00000	0.00001	0.00001	0.00000	0.00002
3	B	0.00000	0.00000	0.00021	0.00000	0.00001	0.00000	0.00022
	J	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	B	0.00095	0.00170	0.00153	0.00142	0.00142	0.00322	0.01024
	J	0.00000	0.00000	0.00001	0.00000	0.00000	0.00000	0.00001

As depicted in figure 4.6, luminescence in the cell is negligible between zero and twelve hours, because population density is low. A quorum is apparently reached slightly after twelve hours, because the concentration of LuxA/B begins to rapidly increase. In this way, the ‘switch’ that controls luminescence is activated when a quorum is achieved.

“Along with LuxR and LuxI, cAMP receptor protein (CRP) plays an important role in controlling luminescence [1:8].” LuxR binds to Ai that forms the complex Co. This complex subsequently binds to the lux box and provides positive feedback for the synthesis of LuxI and negative feedback for the inhibition of LuxR. Increasing levels of LuxI enhance the production of Ai. CRP also binds to the lux box and provides positive feedback for the synthesis of LuxR and negative feedback for the inhibition of LuxI (see figure 3.4).

Figures 4.7, 4.8, and 4.9 illustrate numerical simulations on the time interval—[0 40] hours—of the quorum sensing system that controls luminescence. Initial conditions for these simulations are listed in table 4.10—for trials 1, 2, and 3. These initial conditions are configured in such a way that all species concentrations are identical, with the exception that the concentration for CRP (the factor), which remains constant, but varies from trial to trial. For trials 1–3, CRP is assigned the initial concentrations of 0.7, 20, and 300 nM (nanomolar), respectively.

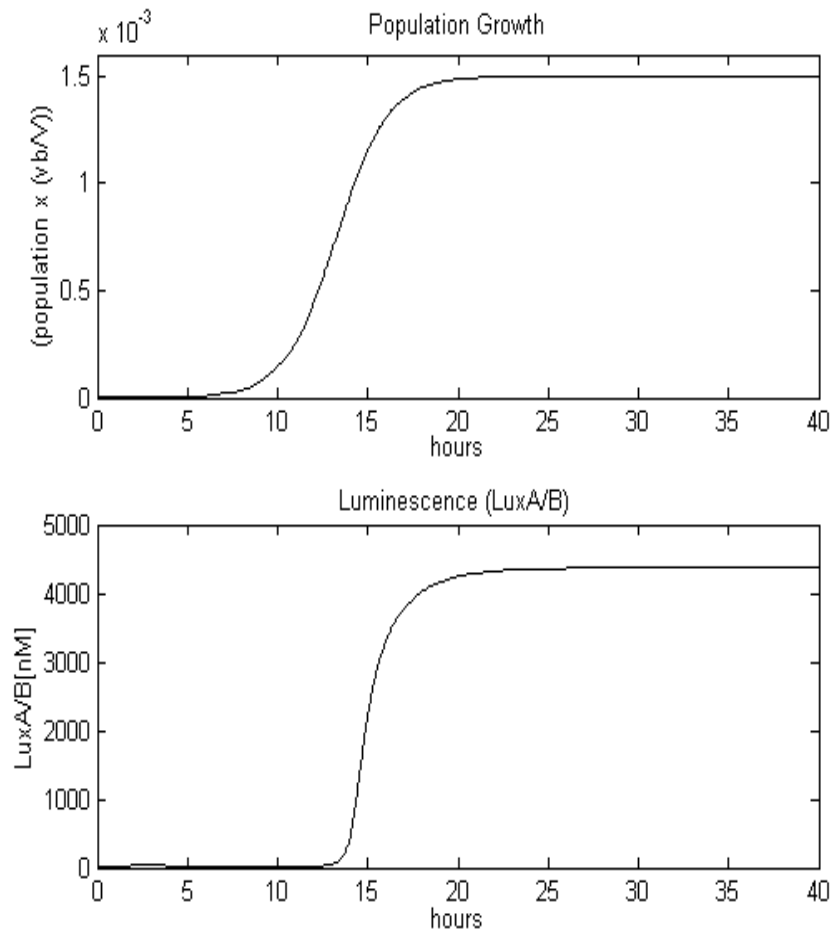


Figure 4.6. The top figure represents time evolution of scaled population for *V. fischeri*: $V_b = 1.5e - 15$ liter and $V = 1.0e-3$ liter. The bottom figure represents time evolution of LuxA/B. The concentration of this protein directly correlates to the intensity of cell luminescence.

The following qualitative characteristics are common for all three simulations, with varying CRP concentrations. At low population densities, there is a short adjustment period (i.e., rapid increase) for proteins, complex Co, and Ai. The concentrations for these species remain nearly constant from 3 to 12 hours. Since the population density is low, luminescence is low (indicated by LuxA/B). Shortly after 12 hours (for trials 2 and 3), another adjustment is initiated, where there is another rapid increase in proteins, Co, and autoinducer Ai. After 20 hours, the concentration for these species approach nearly constant levels. Luminescence in the cell is at its maximum level.

Figures 4.7, 4.8, and 4.9 also illustrate how varying concentrations of CRP (i.e., CRP at low, moderate, and high concentrations) possibly affect luminescence in *V. fischeri*. CRP binds to the lux box and provides positive feedback to O_L (luxR) and negative feedback to O_R (luxICDABEG). The concentration of CRP determines the degree of positive and negative feedback asserted to O_L and O_R .

When the concentration of CRP equals 20, increasing Co provides positive feedback to O_R . This causes the production rate of LuxI and Ai to increase. As the concentration levels of these species increase, the total luminescence increases (i.e., LuxA/B increases). In this case, cell luminescence is high. Figure 4.8 illustrates this behavior.

When the concentration of CRP equals 300, a high degree of negative feedback is applied to O_R . This negative feedback delays the synthesis of LuxI and Ai. Therefore, overall luminescence is moderate. Figure 4.9 illustrates this behavior.

When the concentration of CRP equals 0.7, positive feedback to O_L is low. As a result, a significantly lower concentration of LuxR is initially produced. Consequently, the concentration level of Co is not sufficient to activate O_R . In this case, cell luminescence is extremely low. Figure 4.7 illustrates this behavior.

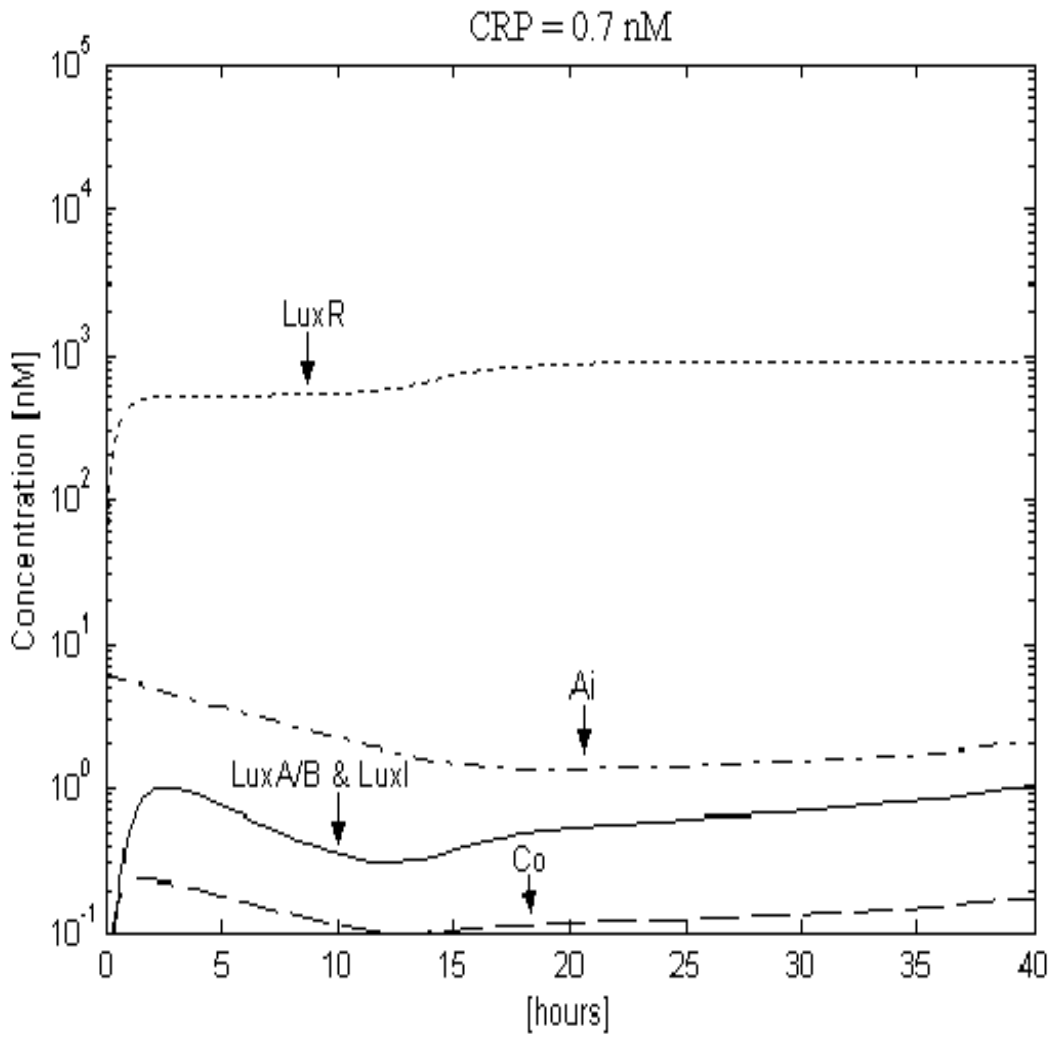


Figure 4.7. This illustrates a numerical simulation (log-linear plot) for LuxR, LuxA/B, LuxI, Ai, and Co in the bioluminescence control model for *V. fischeri*. In this case, CRP = 0.7 (trial 1).

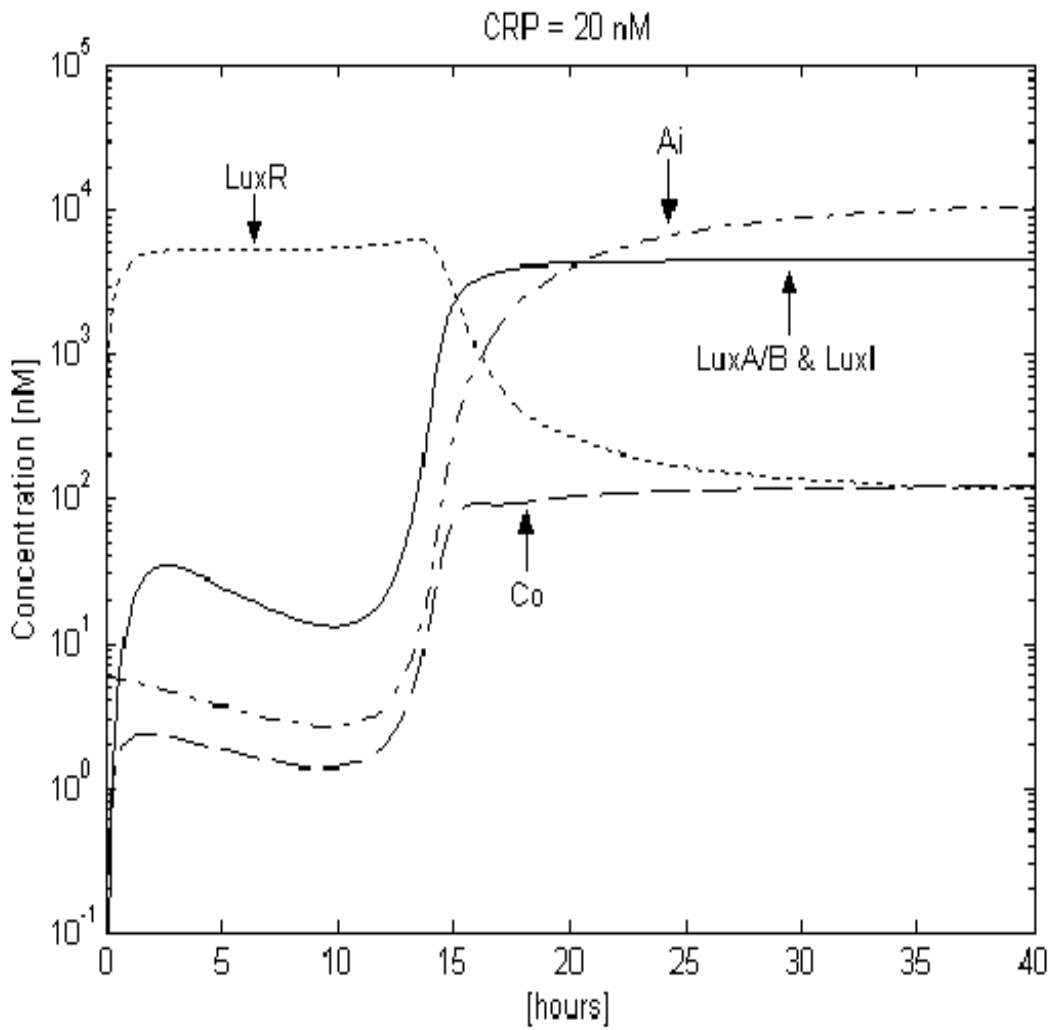


Figure 4.8. This illustrates a numerical simulation (log-linear plot) for LuxR, LuxA/B, LuxI, Ai, and Co in the bioluminescence control model for *V. fischeri*. In this case, CRP = 20 (trial 2).

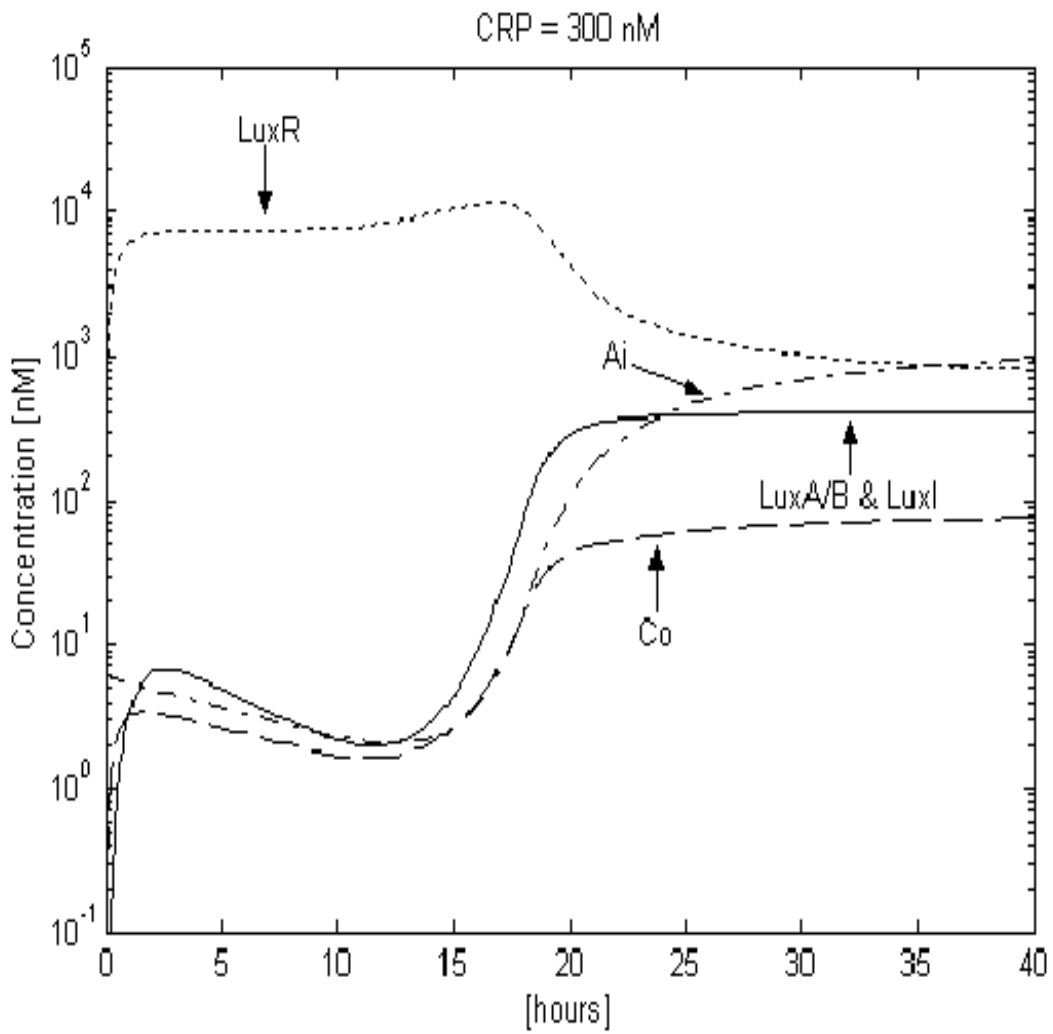


Figure 4.9. This illustrates a numerical simulation (log-linear plot) for LuxR, LuxA/B, LuxI, Ai, and Co in the for *V. fischeri*. In this case, CRP = 300 (trial 3).

4.5.2 Comparison Results. I compare simulation results—for the bioluminescence control model—generated by MATLAB and JigCell. The metrics are absolute error and relative error. All species concentrations are in nano molar concentrations (nM), except for population. The unit of measure for population is the scaled volume of bacteria ($population \times v_b / V$, where $v_b = 1.5\text{E-}15$ liter and $V = 1.0\text{E-}3$ liter [4:4–5]). A set of six simulations are performed on JigCell and MATLAB.

Initial concentrations for all species—separated by trial—are listed in table 4.10. The initial concentrations for each trial are identical, with the exception of CRP. For trials one and four, the concentration of CRP is 0.7. For trials two and five, the concentration of CRP is 20. For trials three and six, the concentration of CRP is 300.

Simulation results performed on the time interval—[0 5] hours—are listed in tables 4.11, 4.12, and 4.13. Simulation results performed on the time interval—[20 25] hours—are listed in tables 4.14, 4.15, and 4.16. With the exception of trial 5 (listed in table 4.15), the largest absolute error between MATLAB and JigCell occurs for species LuxR. During trial 5, the largest absolute error occurs for species Ai. This is not surprising because Ai is significantly larger than all other species.

On the other hand, the relative error is minimal for all trials. The maximum relative error occurs during trials 1 and 3 for scaled population density. Nonetheless, the maximum error is only 0.559%.

4.6 Glycolytic Pathway

4.6.1 General Results. As discussed in section 3.7.4, each of the 10 reactions or steps in the glycolytic pathway (illustrated in figure A) is catalyzed by a specific enzyme. In order for a reaction to take place at each step, a specified enzyme must be present to interact with the substrate. If the required enzyme is absent at a certain step, that reaction is not able to take place. I incorporate this idea into the design of experimental trials conducted in this section.

Table 4.10. Initial Conditions (*V. fischeri*)

Species	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6
population	1.5E-07	1.5E-07	1.5E-07	0.00149	0.00149	0.00149
luxR	0	0	0	0.98132	0.37028	2.58986
luxICDABEG	0	0	0	0.00063	4.81455	0.39765
LuxR	0	0	0	869.16824	265.33072	4241.58358
LuxI	0	0	0	0.54098	4242.81014	285.69095
LuxA/B	0	0	0	0.54098	4242.81014	285.69095
LuxC/D/E	0	0	0	0.09353	720.96687	58.80923
Ai	6	6	6	1.38029	3957.78272	104.74681
C0	0	0	0	0.11666	102.06949	43.03559
CRP	0.7	20	300	0.7	20	300

Table 4.11. Trial 1 Simulation Results (*V. fischeri*)

Species	MATLAB	JigCell	absolute error	relative error
population	4.80855E-06	4.78166E-06	0.00000	0.00559
luxR	0.97021	0.97021	0.00000	0.00000
luxICDABEG	0.00127	0.00127	0.00000	0.00000
LuxR	515.95140	515.93890	0.01250	0.00002
LuxI	0.75306	0.75297	0.00009	0.00012
LuxA/B	0.75306	0.75297	0.00009	0.00012
LuxC/D/E	0.17575	0.17577	0.00002	0.00011
Ai	3.64003	3.64009	0.00006	0.00002
C0	0.17985	0.17985	0.00000	0.00000

Table 4.12. Trial 2 Simulation Results (*V. fischeri*)

Species	MATLAB	JigCell	absolute error	relative error
population	4.80875E-06	4.78166E-06	0.00000	0.00563
luxR	9.80402	9.80402	0.00000	0.00000
luxICDABEG	0.04130	0.04130	0.00000	0.00000
LuxR	5208.60414	5208.48680	0.11734	0.00002
LuxI	24.55205	24.54652	0.00553	0.00023
LuxA/B	24.55205	24.54652	0.00553	0.00023
LuxC/D/E	5.71036	5.7101	0.00022	0.00004
Ai	3.66911	3.66896	0.00015	0.00004
C0	1.82981	1.82969	0.00012	0.00007

Table 4.13. Trial 3 Simulation Results (*V. fischeri*)

Species	MATLAB	JigCell	absolute error	relative error
population	4.81013E-06	4.78166E-06	0.00000	0.00592
luxR	14.10938	14.10937	0.00001	0.00000
luxICDABEG	0.00820	0.00820	0.00000	0.00000
LuxR	7488.10943	7487.93700	0.17243	0.00002
LuxI	4.87340	4.87261	0.00079	0.00016
LuxA/B	4.87340	4.87261	0.00079	0.00016
LuxC/D/E	1.13434	1.13441	0.00007	0.00006
Ai	3.64504	3.64506	0.00002	0.00001
C0	2.61356	2.61350	0.00006	0.00002

Table 4.14. Trial 4 Simulation Results (*V. fischeri*)

Species	MATLAB	JigCell	absolute error	relative error
population	0.00150	0.00150	0.00000	0.00000
luxR	0.98142	0.98142	0.00000	0.00000
luxICDABEG	0.00070	0.00070	0.00000	0.00000
LuxR	882.73431	882.72736	0.00695	0.00001
LuxI	0.61346	0.61345	0.00001	0.00002
LuxA/B	0.61346	0.61345	0.00001	0.00002
LuxC/D/E	0.10408	0.10408	0.00000	0.00000
Ai	1.45469	1.45469	0.00000	0.00000
C0	0.12489	0.12489	0.00000	0.00000

Table 4.15. Trial 5 Simulation Results (*V. fischeri*)

Species	MATLAB	JigCell	absolute error	relative error
population	0.00150	0.00150	0.00000	0.00000
luxR	0.30227	0.30224	0.00003	0.00010
luxICDABEG	4.84908	4.84909	0.00001	0.00000
LuxR	166.69868	166.70787	0.00919	0.00006
LuxI	4358.01106	4357.98780	0.02326	0.00001
LuxA/B	4358.01106	4357.98780	0.02326	0.00001
LuxC/D/E	727.25376	727.25336	0.00040	0.00000
Ai	6989.94902	6989.86470	0.08432	0.00001
C0	113.33358	113.34041	0.00683	0.00006

Table 4.16. Trial 6 Simulation Results (V. fischeri)

Species	MATLAB	JigCell	absolute error	relative error
population	0.00150	0.00150	0.00000	0.00000
luxR	1.45400	1.45381	0.00019	0.00013
luxICDABEG	0.43556	0.43556	0.00000	0.00000
LuxR	1378.32756	1378.45730	0.12974	0.00009
LuxI	387.46805	387.46323	0.00482	0.00001
LuxA/B	387.46805	387.46323	0.00482	0.00001
LuxC/D/E	65.24738	65.24728	0.00010	0.00000
Ai	447.74904	447.74402	0.00502	0.00001
C0	59.98567	59.99049	0.00482	0.00008

In order to present nontrivial results, pyruvate kinase (y_{35}) is ‘removed’ in order to nullify reaction ten. If reaction ten—a one-way reaction—is permitted to take place, the concentrations of all species located upstream from reaction ten would be eventually depleted. Since intermediate reactions five, six, seven, and nine are two-way reactions, it is expected that those species will approach non-zero values as the system evolves towards a critical point. Therefore, I implement a computational model that contains the first 34 equations listed in equation 3.68.

As discussed in section 4.1, the computational cost to obtain accurate simulation results from MATLAB is substantially less than from BioCharon and JigCell. Therefore, simulations for glycolytic pathway are performed only using MATLAB.

BioCharon generates the same system of ODEs (computational model) listed as equation 3.68. However, JigCell generates a reduced computational model, consisting of 16 dependent species and 18 independent species (state variables). This occurs because JigCell automatically identifies conservation relations among species. Conservation relations automatically generated by JigCell are listed in equation 4.1 (T_i is a constant determined by the initial conditions of corresponding species).

Figure 4.10 illustrates the conservation relation listed in equation 4.1 (expression in which y_{13} is the dependent variable). The concentrations of the relevant species at each time step are obtained by performing a simulation of the glycolytic

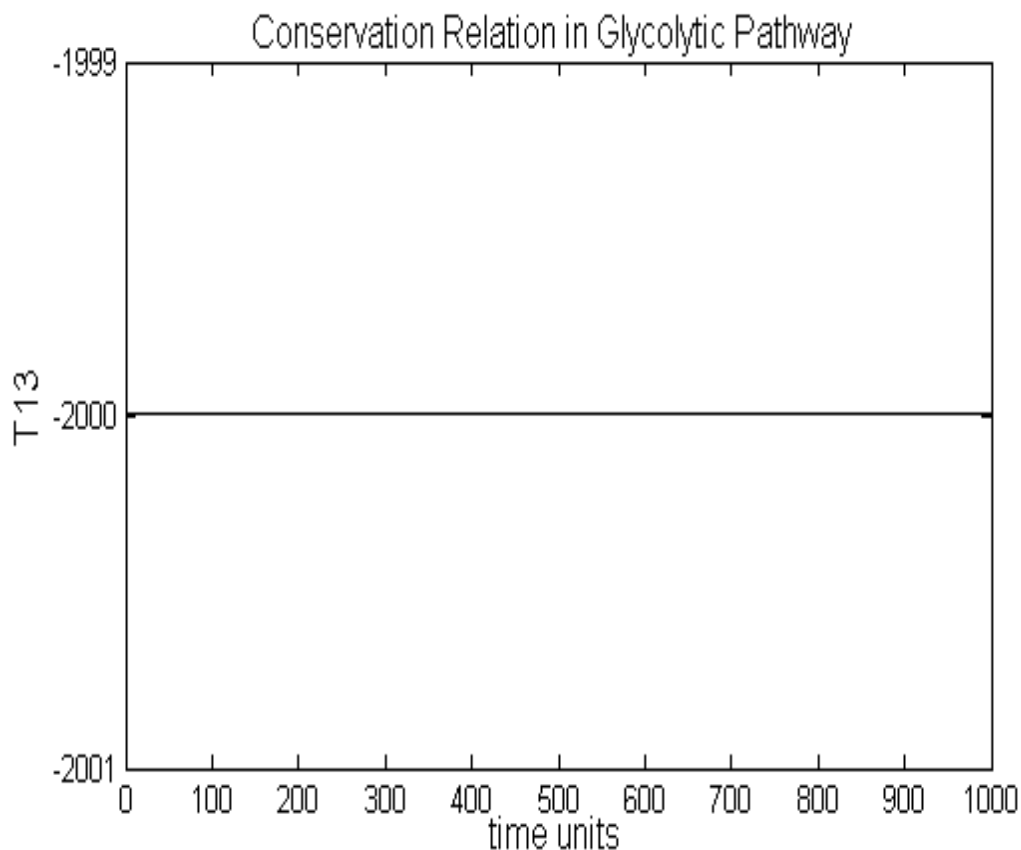


Figure 4.10. This illustrates the conservation relation identified by JigCell, listed in equation 4.1. This is the expression in which y_{13} is the dependent variable. In this case the algebraic expression of evolving species remains constant at -2000 (identified as T_{13} in table 4.18).

pathway, using initial concentrations listed in table 4.17. This plot validates that JigCell correctly identified a relatively complicated conservation relation, consisting of 16 species. In this case, the conservation constant among those species is $T_{13} = -2000$. All relation constants are listed in table 4.18. The values for these constants are derived from the initial species concentrations substituted into the appropriate expression in equation 4.1.

$$y_1 = (T_9 + 0.5 * y_2 + 0.5 * y_5 + 0.5 * y_8 - 0.5 * y_{10} - 0.5 * y_{28} -$$

$$\begin{aligned}
& 0.5 * y_{30} + 0.5 * y_{32} - 0.5 * y_{34} - 0.5 * y_7 - 0.5 * y_{31}) \\
y_6 &= (T_{16} - y_5) \\
y_9 &= (T_{10} - y_8) \\
y_{12} &= (T_4 - y_{11}) \\
y_{13} &= (T_{13} - 0.5 * y_2 + 0.5 * y_5 + 0.5 * y_8 - 0.5 * y_{10} - y_{12} + y_{14} - \\
& 0.5 * y_{16} + 0.5 * y_{18} + 0.5 * y_{20} + 0.5 * y_{28} + 0.5 * y_{16} + \\
& 0.5 * y_{16} + 0.5 * y_{34} - 0.5 * y_7 + 0.5 * y_{31} - 0.5 * y_{17}) \\
y_{15} &= (T_5 - y_{14}) \\
y_{19} &= (T_6 - y_{18}) \\
y_{21} &= (T_{15} + y_{20}) \\
y_{22} &= (T_1 - y_{20} + y_{23}) \\
y_{24} &= (T_7 - y_{23}) \\
y_{25} &= (T_{11} + y_2 - y_5 + y_3 + y_{12} - y_{20} + y_{23} - \\
& y_{28} - y_{30} + y_{32} - y_{34} - y_{31}) \\
y_{26} &= (T_2 + y_2 - y_5 + y_3 + y_{12}) \\
y_{27} &= (T_8 - y_2 + y_5 - y_3 - y_{12}) \\
y_{30} &= (T_{14} - y_{29}) \\
y_{32} &= -(T_3 + y_2 + y_3 - y_4) \\
y_{33} &= (T_{12} - y_{32}) \tag{4.1}
\end{aligned}$$

4.6.2 Comparison Results. I compare simulation results between the computational model generated by JigCell (consisting of 18 state variables) and the computational model listed as equation 3.68 (consisting of 34 state variables). Table 4.17 lists initial species concentrations for simulation results listed in table 4.19. Simulation results from the full and reduced models are listed in columns one and two of table 4.19. Absolute error is listed in column three. The unit of measure for

species concentrations can be in any consistent unit. The simulation time is 1000 seconds.

The maximum absolute error occurs for species y_2 (listed in table 4.19). This error is negligible, because the error is just 0.0003%. Therefore, the reduced model is validated by MATLAB results.

Table 4.17. Initial Conditions (Glycolytic Pathway)

Species	[Value]	Species	[Value]	Species	[Value]	Species	[Value]
y_1	500	y_2	1000	y_3	1000	y_4	1000
y_5	1000	y_6	0.00001	y_7	0.00001	y_8	1000
y_9	0.00001	y_{10}	0.00001	y_{11}	1000	y_{12}	0.00001
y_{13}	0.00001	y_{14}	1000	y_{15}	0.00001	y_{16}	0.00001
y_{17}	0.00001	y_{18}	1000	y_{19}	0.00001	y_{20}	1000
y_{21}	1000	y_{22}	0.00001	y_{23}	1000	y_{24}	0.00001
y_{25}	0.00001	y_{26}	1000	y_{27}	0.00001	y_{28}	0.00001
y_{29}	1000	y_{30}	0.00001	y_{31}	0.00001	y_{32}	1000
y_{33}	0.00001	y_{34}	0.00001				

4.7 Yeast Cell Cycle

4.7.1 General Results. As discussed in section 3.7.3, the cell division cycle of budding yeast is characterized by the system alternating between two self-maintaining states— $State_1$ and $State_2$. CDK activity (combined activities of Clb’s and Cln’s) is low in $State_1$, while their enemies—Sic1 and Hct1 are high. Therefore, concentration levels of the Clb’s and Cln’s remain mostly low during $State_1$. On the other hand, CDK activity is high in $State_2$. As a result, the concentrations of the Clb’s and Cln’s are high, where their enemies’ concentrations remain mostly low during $State_2$.

Control is transferred from $State_1$ to $State_2$ (and vice versa) by means of two transitions—**Start** and **Finish**. In the molecular control model of CDK activities (figure 3.5), **Start** is initiated when the concentration of species ORI reaches a concentration of 1 unit (approached from below) [1:374]. **Finish** is initiated when species

Table 4.18. Conservation Relation Constants (Glycolytic Pathway)

Constant	Value
T_1	0.00001
T_2	-0.00001
T_3	-2000
T_4	1000.000010
T_5	1000.000010
T_6	1000.000010
T_7	1000.00001
T_8	1000.000020
T_9	-1499.999970
T_{10}	1000.000010
T_{11}	-1999.999960
T_{12}	1000.000010
T_{13}	-1999.999980
T_{14}	1000.000010
T_{15}	0
T_{16}	1000.000010

SPN reaches a concentration of 1 unit (approached from below) [1:374]. Simulation results, generated by MATLAB from equations 3.66 and 3.67, depict important characteristics of molecular interaction in the budding yeast cell division cycle. These results are illustrated in figures 4.11, 4.12, and 4.13.

Activities of Clb2T, Clb5T, and Cln2 are illustrated in figure 4.11. Collectively, the activities of these species are low during $State_1$ and high during $State_2$. Activities of Sic1T and Hic1T (enemies of CDKs) are illustrated in figure 4.12. Their activities are opposite from the CDKs. Figure 4.13 illustrates cell division. Cell division occurs around 380 minutes. Rules for cell division are listed in the caption of figure 4.13.

4.7.2 Comparison Results. In the previous sections, initial species concentrations are perturbed during each trial. In this section, the reaction rate constant, $k_{d,20}$, is perturbed from a value of 0.080 to 0.092 at 0.006 increments. This rate constant is listed in equation 3.66 (in the differential equation defining the rate of change for species Cdc20). Initial species concentrations and other model param-

Table 4.19. Comparison Results (Glycolytic Pathway)

Species	Full Model	Reduced Model	Absolute Error
	1.0E+003 *	1.0E+003 *	1.0E-005 *
y_1	0.0000000000000000	0.0000000000000001	0.00000112688929
y_2	0.32719541332651	0.32719541606607	0.27395537358643
y_3	1.01346976779097	1.01346976794359	0.01526191226731
y_4	0.01673690254825	0.01673690290803	0.03597766031760
y_5	1.00000001000000	1.00000000999480	0.00052004907047
y_6	0.0000000000000000	0.000000000000520	0.00052003770179
y_7	0.0000000000000000	0.0000000000000001	0.00000067220655
y_8	1.00000001000000	1.00000000999543	0.00045685055738
y_9	0.0000000000000000	0.000000000000457	0.00045682782002
y_{10}	0.0000000000000000	0.0000000000000000	0.00000000010772
y_{11}	1.00000001000000	1.00000000999601	0.00039946144170
y_{12}	0.0000000000000000	0.000000000000399	0.00039934775486
y_{13}	0.00000000013785	0.00000000013526	0.00025951412943
y_{14}	0.99999959644638	0.99999959644794	0.00015545538190
y_{15}	0.00000041355362	0.00000041355206	0.00015550629033
y_{16}	0.00000334154936	0.00000334154891	0.00004481718911
y_{17}	0.00000037128326	0.00000037128321	0.00000510898911
y_{18}	0.99888739948217	0.99888739963078	0.01486117753302
y_{19}	0.00111261051783	0.00111261036922	0.01486108365256
y_{20}	0.00111698053804	0.00111698060354	0.00654993137505
y_{21}	0.00111698053804	0.00111698060354	0.00654993135285
y_{22}	0.98653033025512	0.98653033007102	0.01840957111199
y_{23}	0.98764730047205	0.98764730067456	0.02025145704465
y_{24}	0.01235270952795	0.01235270932544	0.02025145473539
y_{25}	0.00000003803393	0.00000003803361	0.00003179414129
y_{26}	0.34066516520734	0.34066516401886	0.11884819173247
y_{27}	0.65933484479266	0.65933484598114	0.11884818604813
y_{28}	0.00000106027192	0.00000106027190	0.00000260618866
y_{29}	0.99682927970533	0.99682927978787	0.00825422148409
y_{30}	0.00317073029467	0.00317073021213	0.00825437611596
y_{31}	0.00000954244732	0.00000954244706	0.00002542509883
y_{32}	0.67607171776758	0.67607171889837	0.11307893146295
y_{33}	0.32392829223242	0.32392829110163	0.11307886893519
y_{34}	0.00008588202584	0.00008588202355	0.00022895867979

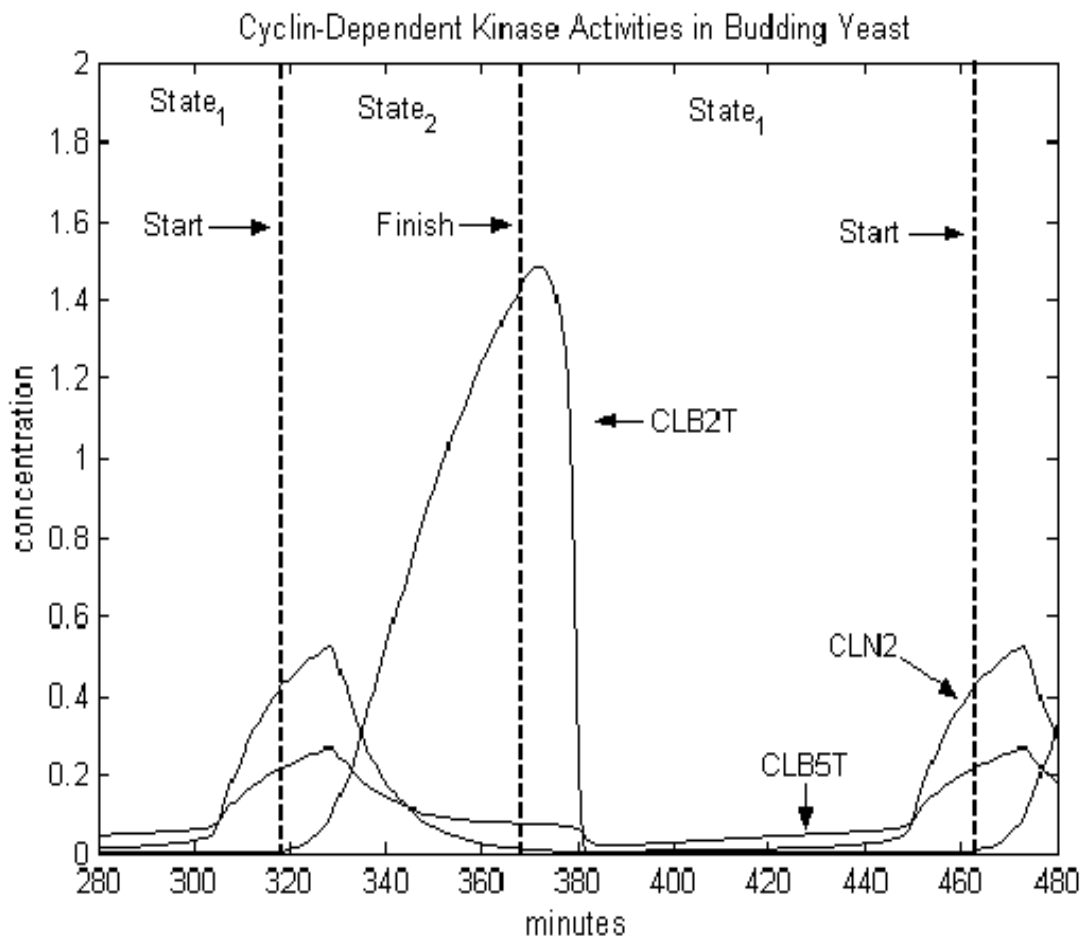


Figure 4.11. This depicts the activity levels of Clb's and Cln's. During *State*₁, the activity of both cyclins are low. However, their activity is high during *State*₂.

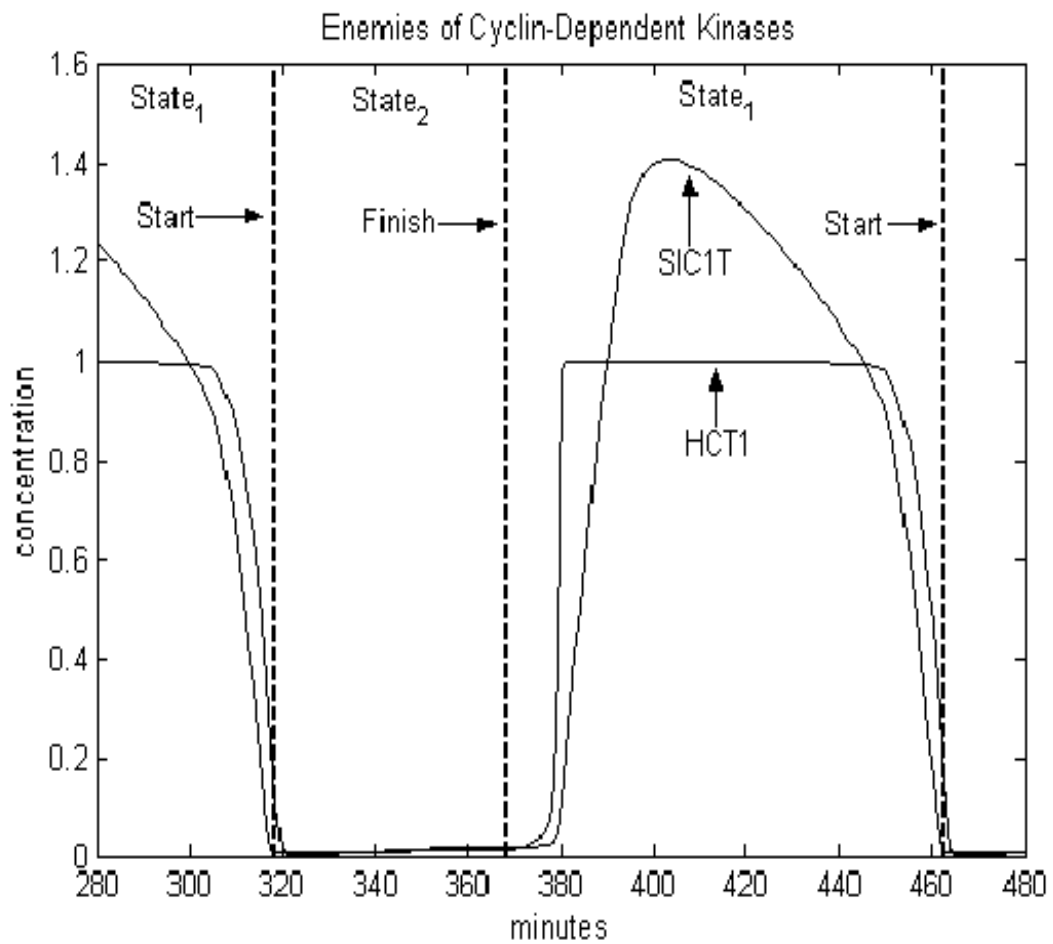


Figure 4.12. This depicts the activity levels of Sic1 and HCT1—enemies of Clb's. Their activity levels are opposite of the Clb's and Cln's, displayed in figure 4.11.

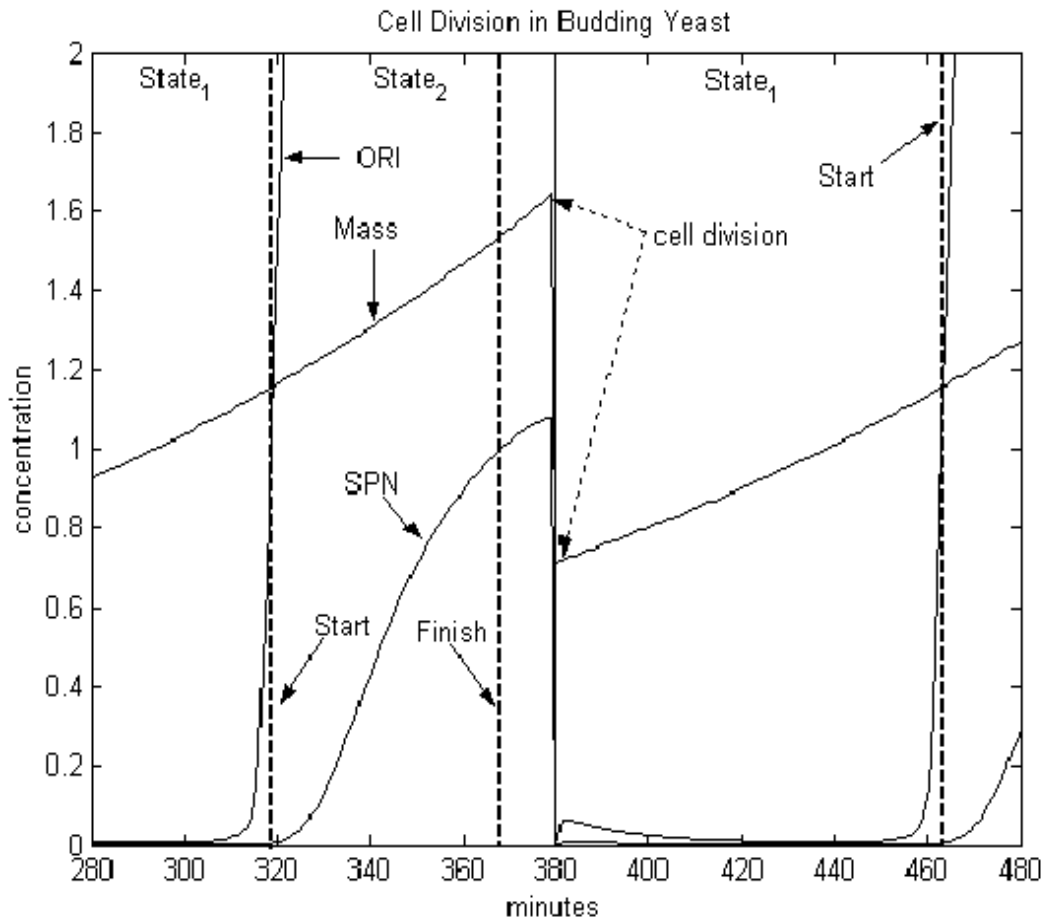


Figure 4.13. Cell division occurs around 380 minutes. Cell division is initiated when the concentration of Clb2 drops below a threshold of 0.3 concentration units. At cell division, the mass is divided between the mother and daughter cell as follows:
 (mass of daughter cell at birth) = $0.433 \times$ (mass at cell separation)
 and (mass of mother cell at birth) = $(1 - 0.433) \times$ (mass at cell separation) [8:374].

ters remain constant for all three trials (values are listed in appendix A—xpp output file).

The metrics for comparison results are absolute error and sum of absolute error (between MATLAB and JigCell simulation results). “All concentration variables are scaled so that their maximal values are pure numbers of order 1 [1:386].” The sampled time interval is [280 480] minutes.

Tables 4.20 and 4.21 list the percentage of change caused by perturbing $k_{d,20}$. In both tables listing simulation results generated by JigCell and MATLAB, respectively, $Sic1_T$ is the most sensitive parameter. $Sic1_T$ is decreased -18.1% on JigCell and decreased -17.1% in MATLAB.

Comparison results (between JigCell and MATLAB) are listed in table 4.22. The length parameter is the interval immediately after cell division and right before cell division occurs. For all trials, maximum absolute error occurs in Length. The maximum sum of absolute error occurs during trial 3.

Table 4.20. MATLAB $k_{d,20}$ Perturbation Results (Budding Yeast)

Name	Trial 1	Trial 2	Trial 3	Overall Change
$Clb2_T$	1.48672	1.52294	1.54121	3.7%
$Clb5_T$	0.26888	0.26682	0.26644	-0.9%
$Cln2$	0.52380	0.51813	0.51442	-1.8%
$Sic1_T$	1.40809	1.29039	1.15385	-18.1%
$Hct1$	0.99995	0.99995	0.99989	0.0%
Length	145.50000	144.10000	143.79830	-1.2%
$k_{d,20}$	0.080	0.086	0.092	15.0%

4.8 Summary

In this chapter I perform numerical simulations for the following models: modified Brusselator, lysis-lysogeny pathway of a mutant bacteriophage, quorum sensing system that controls luminescence in *V. fischeri*, a glycolytic pathway, and control system for CDKs in budding yeast cell division cycle. I present general results

Table 4.21. JigCell $k_{d,20}$ Perturbation Results (Budding Yeast)

Name	Trial 1	Trial 2	Trial 3	Overall Change
$Clb2_T$	1.48167	1.50582	1.52887	3.2%
$Clb5_T$	0.26897	0.26767	0.26654	-0.9%
$Cln2$	0.52345	0.51816	0.51367	-1.9%
Sic_T	1.39631	1.26793	1.15715	-17.1%
$Hct1$	0.99986	0.99985	0.99985	0.0%
Length	144.90000	144.90000	144.90000	0.0%
$k_{d,20}$	0.080	0.086	0.092	15.0%

Table 4.22. Comparison Results (Budding Yeast)

Name	Trial 1	Trial 2	Trial 3
$Clb2_T$	0.00505	0.01712	0.01234
$Clb5_T$	0.00009	0.00085	0.00010
$Cln2$	0.00035	0.00003	0.00075
Sic_T	0.01178	0.02246	0.00330
$Hct1$	0.00009	0.00010	0.00004
Length	0.60000	0.80000	1.10170
Total Error	0.61735	0.84055	1.11824

that depict important characteristics of each model. In order to perform a quasi-performance test for BioCharon and Jigcell, I compare simulation results generated by these software tools.

The models for *V. fischeri* and budding yeast are based on journal publications (references [1], [4], [8]). I successfully duplicate the published results.

V. Summary and Conclusion

5.1 Summary

The overall goal of this thesis was to implement mathematical models (via rate-equation approach) for three well-documented cellular systems using BioCharon, JigCell, and MATLAB. Each model characterizes a specific control mechanism that synchronizes a set of intracellular processes within a genetic regulatory network. The experience gained from implementing these models is intended to contribute to Air Force toxicology studies for the following reason. “Mathematical models are useful for providing a framework for integrating data and gaining insights into the static and dynamic behavior of complex biological systems such as networks of interacting genes [26:247].” In short, the augmentation of intracellular models into Air Force toxicology studies has the potential to significantly improve the quality and efficiency of the studies.

In order to achieve my overall goal, I completed the following tasks.

1. Acquired a sound understanding of the fundamentals of biochemistry and microbiology that are related to biological reaction systems.
2. Implemented computational (mathematical) models via rate-equation approach for the following molecular systems:
 - modified Brusselator
 - glycolytic pathway
 - lysis-lysogeny pathway of bacteriophage
 - bioluminescence control in *V. fischeri*
 - cell cycle control in budding yeast
3. Became proficient in using three software tools: BioCharon, JigCell, and MATLAB.

In addition, simulation results generated by BioCharon and JigCell were compared to simulation results generated by MATLAB to gauge the accuracy of these two trial releases by DARPA.

5.2 Conclusion

Reaching the overall goal of this thesis has led to a number of conclusions. The rate-equation approach is appropriate for deterministically modeling both small and large molecular systems (e.g., the modified Brusselator and the glycolytic pathway). There is a clear need for the use of software modeling tools that automate all or some of the steps in the rate-equation approach. Without such tools, the modeler easily becomes overburdened in attempting to derive a computational model from other than a simplistic biological system.

From this thesis, the merits of BioCharon and JigCell are apparent. The graphical approach—implemented by BioCharon—is highly intuitive and represents the higher level of abstraction in building models. By means of a GUI, the user is able to quickly create a reaction system by simply manipulating two main objects (nodes and arcs). The spreadsheet approach—implemented by JigCell—is relatively user-friendly; however, the user must enter the entire set of reaction equations to characterize a reaction system. Both software tools allow the user to view the computational model generated from user input. This is crucial in the debugging process.

The three primary cellular models—lysis-lysogeny pathway, bioluminescence control, and cell cycle control—are outstanding learning aids in comprehending gene expression (when a gene is active during transcription). Gene expression is represented at various levels of detail in each model. For the lysis-lysogeny pathway model, transcription and translation are combined into equation 3.41. The left hand side of equation 3.41 depicts transcription of mRNA transcripts, and the right hand side depicts translation of mRNA transcripts into n protein molecules. For the bioluminescence control model, the rates for synthesis of mRNA transcripts from

the left and right operons (O_L and O_R) are depicted by \dot{x}_1 and \dot{x}_2 in equation 3.62. Rates for translation of messenger RNA's into proteins are depicted by \dot{x}_3 , \dot{x}_4 , \dot{x}_5 , and \dot{x}_6 in the same equation. For the cell cycle control model, transcription and translation rates are collectively modulated by transcription factors (e.g., the activation level of transcription factor SBF enhances the synthesis of Cln2 in figure 3.5).

Gene expression is directly relevant to toxicology studies. "Almost without exception, gene expression is altered during toxicity, as either a direct or indirect result of toxicant exposure. The challenge facing toxicologists is to define, under a given set of experimental conditions, the characteristic and specific pattern of gene expression elicited by a given toxicant [21:153]." Since gene expression can be characterized by mathematical models, specific models can be used to give predictions on how toxicant exposures affect cells.

5.3 Recommendations

Further research into the construction and evaluation of intracellular models would benefit Air Force toxicology studies. In gaining the necessary experience, five reaction systems were thoroughly studied and implemented. The next step could be to construct a model that is coupled with experimental data (previously discussed section 2.1). Such data could be freely obtained from the public domain and/or from the Air Force.

The use of deterministic models to depict reaction systems is not always appropriate. "Conventional deterministic kinetics cannot be used to predict statistics of regulatory systems that produce probabilistic outcomes. Rather, a stochastic kinetic analysis must be used to predict statistics of regulatory outcomes for such stochastically regulated systems [3:1633]." Therefore, it is worthwhile to thoroughly study and implement mathematical models using the stochastic kinetics approach (previously discussed in section 2.8).

Even though BioCharon and JigCell have user-friendly front ends for modeling reaction systems, their respective ODE solvers need improvement. The computational cost for both tools was larger than that for MATLAB when attempts were made to implement models for the bioluminescence control and glycolytic pathway models. Once an ODE solver that is comparable to MATLAB's accuracy and efficiency is incorporated into both software packages, both will become very desirable tools for modeling and analyzing deterministic systems.

Appendix A. JigCell Model Builder Entries

In this section, I present my entries for each *model* spreadsheet (using JigCell Model Builder) and the corresponding output files (full xpp) that characterize the following models: lysis-lysogeny pathway, quorum sensing system, and a control system for the budding yeast cell cycle. The spreadsheets are illustrated in the following figures.

Figure A.1: lysis-lysogeny pathway

Figure A.2: quorum sensing system

Figure A.3: control system for budding yeast cell cycle

Each output file is listed immediately after its corresponding *model* spreadsheet.

I only list entries for the *Reaction* and *Modifiers and Contants* columns from each spreadsheet. The user enters the set of reaction equations in the *Reaction* column. In the *Modifiers and Contants* column, the user defines each rate law.

An important aspect of user interaction with the *model* spreadsheet is that the user cannot explicitly define a reversible reaction. Consequently, the user must divide a reversible reaction into two separate, one-way reactions. For example, the following two reaction equations are the entries for $D + X_2 \xrightleftharpoons[k_{-2}]{k_2} DX_2$ (equation 3.38).

1. $D + X_2 \rightarrow DX_2$
2. $DX_2 \rightarrow D + X_2$

Entries (user input is underlined) in the *Modifiers and Contants* column for reactions 1 and 2, respectively, are $K_f = \underline{k_2}$ and $K_f = \underline{k_{-2}}$. The user must assign numerical values to k_2 and k_{-2} in a separate spreadsheet (the *Constants* spreadsheet). Detailed information about JigCell Model Builder can be found in [30].

	Reaction	Modifiers and Constants
1	$\rightarrow X$	$K_f=0.4$
2	$2X \rightarrow X_2$	$K_f=1$
3	$X_2 \rightarrow 2X$	$K_f=1$
4	$D+X_2 \rightarrow DX_2$	$K_f=1$
5	$DX_2 \rightarrow D+X_2$	$K_f=1$
6	$D+X_2 \rightarrow DXS_2$	$K_f=1$
7	$DXS_2 \rightarrow D+X_2$	$K_f=1$
8	$DX_2 + X_2 \rightarrow DX_2X_2$	$K_f=1$
9	$DX_2X_2 \rightarrow DX_2+X_2$	$K_f=1$
10	$DX_2+P \rightarrow DX_2+P+4X$	$K_f=0.8$
11	$X \rightarrow A$	$K_f=6$

Figure A.1. These are my spreadsheet entries that characterize the lysis-lysogeny pathway (previously discussed in section 3.7.1).

```

*****OUTPUT FILE (lysis-lysogeny pathway)*****
# C:\appendix\lysis_lysoyeny.odef  Generated by JigCell
#Functions
#Dependent species
DX2X2=(T1 - DX2 - DXS2 - D)/(-(-1.0))
P=(T2)
#Species
#Independent Species
dX2/dt= - 1*X2 - 1*D*X2 - 1*D*X2 - 1*DX2*X2 + 1*X*X + 1*DX2 + 1*DXS2 + 1*DX2X2
dDX2/dt= - 1*DX2 - 1*DX2*X2 - .8*DX2*P + 1*D*X2 + 1*DX2X2 + .8*DX2*P
dD/dt= - 1*D*X2 - 1*D*X2 + 1*DX2 + 1*DXS2
dDXS2/dt= - 1*DXS2 + 1*D*X2
dX/dt= - 2*1*X*X - 6*X + .4 + 2*1*X2 + 4*.8*DX2*P
dA/dt=6*X
#Globals
#Initial Conditions
init X2=.78934, DX2=1.23268, D=1.56164, DXS2=1.23268
init X=.88845, A=0
#Constants
param T2=1.25, T1=5
#Plot dependent species
aux DX2X2=DX2X2
aux P=P
done
*****OUTPUT FILE (lysis-lysogeny pathway)*****

```

	Reaction	Modifiers and Constants
1	Sig	$A1^*A3/(A2^*A3+A1^*A3)$
2	OpSig	$1-(A1^*A3/(A2^*A3+A1^*A3))$
3	kG	$kg1*(1-(A1/x0max))$
4	->population	$Kf=kG(population)*population$
5	->luxR	$Kf=Tc*(OpSig(C0, Kc0, Vc0)*Sig(CRP, KCRP, VCRP)+b)$
6	luxR->	$Kf=1/HRNA$
7	luxR->	$Kf=kG(population)$
8	->luxICDABEG	$Kf=Tc*(Sig(C0, Kc0, Vc0)*OpSig(CRP, KCRP, VCRP)+b)$
9	luxICDABEG->	$Kf=1/HRNA$
10	luxICDABEG->	$Kf=kG(population)$
11	->LUXR	$Kf=Tl*luxR$
12	LUXR->	$Kf=1/Hsp$
13	LUXR->	$Kf=kG(population)$
14	LUXR+Ai->C0	$Kf=rAIR$
15	C0->LUXR+Ai	$Kf=rC0$
16	->LUXI	$Kf=Tl*luxICDABEG$
17	LUXI->	$Kf=1/Hsp$
18	LUXI->	$Kf=kG(population)$
19	->LUXAB	$Kf=Tl*luxICDABEG$
20	LUXAB->	$Kf=1/Hsp$
21	LUXAB->	$Kf=kG(population)$
22	->LUXCDE	$Kf=Tl*luxICDABEG$
23	LUXCDE->	$Kf=1/Hsp$
24	LUXCDE->	$Kf=kG(population)$
25	->Ai	$Kf=population*(rAII*LUXI -$
	****	$rAIR*Ai*LUXR+rC0*C0)+(rAIR*LUXR*Ai - rC0*C0)$
26	Ai->	$Kf=1/HAI$
27	C0->	$Kf=1/Hsp$
28	C0->	$Kf=kG(population)$

Figure A.2. These are my spreadsheet entries that characterize the quorum sensing system that controls luminescence in *V. fischeri* (previously discussed in section 3.7.2). Functions are defined in lines 1, 2, and 3 by corresponding algebraic expressions listed in the *Modifiers and Constants* column.


```

*****OUTPUT FILE (V. fischeri)*****
# C:\appendix\vfish.odef  Generated by JigCell
#Functions
Sig(A1,A2,A3)=A1^A3/(A2^A3+A1^A3)
OpSig(A1,A2,A3)=1-(A1^A3/(A2^A3+A1^A3))
kG(A1)=kg1*(1-(A1/x0max))
#Dependent species
#Species
#Independent Species
dluxR_18/dt= - 1/HRNA*luxR_18 - kG(popula_26)*luxR_18 +
              (Tc*(OpSig(C0, Kc0, Vc0)*Sig(CRP, KCRP, VCRP)+b))
dLUXR_19/dt= - 1/Hsp*LUXR_19 - kG(popula_26)*LUXR_19 - rAIR*LUXR_19*Ai +
              Tl*luxR_18 + rC0*C0
dC0/dt= - rC0*C0 - 1/Hsp*C0 - kG(popula_26)*C0 + rAIR*LUXR_19*Ai
dLUXAB/dt= - 1/Hsp*LUXAB - kG(popula_26)*LUXAB + Tl*luxICD_25
dAi/dt= - rAIR*LUXR_19*Ai - 1/HAI*Ai + rC0*C0 +
         (popula_26*(rAII*LUXI-rAIR*Ai*LUXR_19+rC0*C0)+
         (rAIR*LUXR_19*Ai - rC0*C0))
dLUXCDE/dt= - 1/Hup*LUXCDE - kG(popula_26)*LUXCDE + Tl*luxICD_25
dLUXI/dt= - 1/Hsp*LUXI - kG(popula_26)*LUXI + Tl*luxICD_25
dluxICD_25/dt= - 1/HRNA*luxICD_25 - kG(popula_26)*luxICD_25 +
               (Tc*(Sig(C0, Kc0, Vc0)*OpSig(CRP, KCRP, VCRP)+b))
dpopula_26/dt=kG(popula_26)*popula_26
#Globals
#Initial Conditions
init luxR_18=0, LUXR_19=0, C0=0, LUXAB=0
init Ai=6, LUXCDE=0, LUXI=0, luxICD_25=0
init popula_26=.00000015

```

```
#Constants
param Kc0=20, CRP=10, VCRP=1, HRNA=60
param Hsp=3600, rC0=.01, Tc=0.25, KCRP=10
param Tl=0.25, b=.00001, rAIR=.000001, Vc0=2
param kg1=.0001925, x0max=.0015, Hup=600, rAII=.05
param HAI=36000
#Plot dependent species
done
*****OUTPUT FILE (V. fischeri)*****
```

	Reaction	Modifiers and Constants
1	->Timer1	Kf=1
2	->Timer2	Kf=(1/12)
3	->Alpha	Kf=0
4	->Beta	Kf=0
5	->CLN2	Kf=(ksn2'+ksn2''*SBF)*MASS
6	CLN2->	Kf=kdn2
7	->CLB2T	Kf=(ksb2'+ksb2''*MCM1)*MASS
8	CLB2T->	Kf=Vdb2
9	Vdb2	kdb2*(HCT1T-HCT1)+kdb2''*HCT1+kdb2'''*CDC20
10	->CLB5T	Kf=(ksb5'+ksb5''*MBF)*MASS
11	CLB5T->	Kf=Vdb5
12	Vdb5	kdb5'+kdb5''*CDC20
13	CLN3S	CLN3MAX*(Dn3*MASS)/(Jn3 + Dn3*MASS)
14	->SIC1T	Kf=ksc1' + ksc1''*SWI5
15	SIC1T->	Kf=(kd1c1+(Vd2c1/(Jd2c1 + SIC1T)))
16	->C2	Kf=kasb2*(CLB2T - C2)*(SIC1T -(C2+C5))
17	C2->	Kf=(kdib2+Vdb2+kd1c1+(Vd2c1/(Jd2c1+SIC1T)))
18	->C5	Kf=kasb5*(CLB5T - C5)*(SIC1T -(C2+C5))
19	C5->	Kf=(kdib5+Vdb5+kd1c1+(Vd2c1/(Jd2c1+SIC1T)))
20	->CDC20T	Kf=(ks20'+ks20''*(CLB2T - C2))
21	CDC20T->	Kf=kd20
22	->CDC20	Kf=ka20*(CDC20T-CDC20)
23	CDC20->	Kf=(Alpha*Vi20+Beta*(-9.9/12)*Timer1+10)+kd20
24	->Vi20	Kf=0
25	->HCT1	Kf=((kat1'+kat1''*CDC20)*(HCT1T-HCT1))/(Jat1+HCT1T-HCT1)
26	HCT1->	k1=1; M1=Vit1; J1=Jit1
27	Vit1	kit1'+kit1''*(CLN3S+eit1n2*CLN2+eit1b5*CLB5+eit1b2*CLB2)
28	Vasbf	kasbf*(CLN2+esbf3*(CLN3S+BCK2)+esbf5*CLB5)
29	SBF	G(Vasbf,kisbf+kisbf''*CLB2,Jasbf,Jisbf)
30	MBF	G(Vasbf,kisbf+kisbf''*CLB2,Jasbf,Jisbf)
31	MCM1	G(kamcm*CLB2,kimcm,Jamcm,Jimcm)
32	Vd2c1	kd2c1*(ec1n3*CLN3S+ec1k2*BCK2+CLN2+ec1b5*CLB5+ec1b2*CLB2)
33	->ORI	Kf=ksori*((CLB5T-C5)+eorib2*(CLB2T - C2))
34	ORI->	Kf=kdori
35	->BUD	Kf=ksbud*(CLN2+CLN3S+ebudb5*(CLB5T - C5))
36	BUD->	Kf=kdbud
37	->SPN	Kf=ksspnp*((CLB2T - C2)/(Jspn+(CLB2T - C2)))
38	SPN->	Kf=kdspn
39	->MASS	Kf=mu*MASS
40	BCK2	BCK20*MASS
41	G	(2*A1*A4)/(A2-A1+A1*A4+A2*A3+sqrt((A2-A1+A1*A4+A2*A3)^2-4*(A2-A1)*A1*A4))
42	SWI5	G(kaswi*CDC20,kiswi+kiswi''*CLB2,Jaswi,Jiswi)
43	CLB2	CLB2T-C2
44	CLB5	CLB5T-C5
45	SIC1	SIC1T-(C2+C5)

Figure A.3. These are my spreadsheet entries that characterize the budding yeast cell cycle (previously discussed in section 3.7.3). The single function is defined in line 41. The remaining entries define state variables.

```

*****OUTPUT FILE (budding yeast)*****
# C:\appendix\yeast.odef  Generated by JigCell
#Functions
G(A1,A2,A3,A4)=(2*A1*A4)/(A2-A1+A1*A4+A2*A3+
((A2-A1+A1*A4+A2*A3)^2-4*(A2-A1)*A1*A4)^.5)
#Dependent species
#Species
Vdb2=(kdb2')*((HCT1T)-HCT1)+(kdb2'')*HCT1+(kdb2''')*CDC20
aux Vdb2=Vdb2
Vdb5=(kdb5')+(kdb5'')*CDC20
aux Vdb5=Vdb5
CLN3S=(CLN3MAX)*((Dn3)*MASS)/((Jn3) + (Dn3)*MASS)
aux CLN3S=CLN3S
Vit1=(kit1')+(kit1'')*(CLN3S+(eit1n2)*CLN2+(eit1b5)*CLB5+(eit1b2)*CLB2)
aux Vit1=Vit1
Vasbf=(kasbf)*(CLN2+(esbfn3)*(CLN3S+BCK2)+(esbfb5)*CLB5)
aux Vasbf=Vasbf
SBF=G(Vasbf,(kisbf')+(kisbf'')*CLB2,(Jasbf),(Jisbf))
aux SBF=SBF
MBF=G(Vasbf,(kisbf')+(kisbf'')*CLB2,(Jasbf),(Jisbf))
aux MBF=MBF
MCM1=G((kamcm)*CLB2,(kimcm),(Jamcm),(Jimcm))
aux MCM1=MCM1
Vd2c1=(kd2c1)*((ec1n3)*CLN3S+(ec1k2)*BCK2+CLN2+(ec1b5)*CLB5+(ec1b2)*CLB2)
aux Vd2c1=Vd2c1
BCK2=(BCK20)*MASS
aux BCK2=BCK2
SWI5=G((kaswi)*CDC20,(kiswi')+(kiswi'')*CLB2,(Jaswi),(Jiswi))

```

```

aux SWI5=SWI5
CLB2=CLB2T-C2
aux CLB2=CLB2
CLB5=CLB5T-C5
aux CLB5=CLB5
SIC1=SIC1T-(C2+C5)
aux SIC1=SIC1
#Independent Species
dCLB2T/dt= - Vdb2*CLB2T + ((ksb2'+ksb2''*MCM1)*MASS)
dSIC1T/dt= - ((kd1c1+(Vd2c1/(Jd2c1 + SIC1T))))*SIC1T + (ksc1' + ksc1''*SWI5)
dC5/dt= - ((kdib5+Vdb5+kd1c1+(Vd2c1/(Jd2c1+SIC1T))))*C5 +
          (kasb5*(CLB5T - C5)*(SIC1T -(C2+C5)))
dCDC20/dt= - ((Alpha*Vi20+Beta*((-9.9/12)*Timer1+10)+kd20))*CDC20 +
          (ka20*(CDC20T-CDC20))
dHCT1/dt= (((kat1'+kat1''*CDC20)*(HCT1T-HCT1))/(Jat1+HCT1T-HCT1)) -
          (1*HCT1*Vit1)/(Jit1+HCT1)
dBUD/dt= - kdbud*BUD + (ksbud*(CLN2+CLN3S+ebudb5*(CLB5T - C5)))
dCLB5T/dt= - Vdb5*CLB5T + ((ksb5' +ksb5''*MBF)*MASS)
dCDC20T/dt= - kd20*CDC20T + ((ks20'+ks20''*(CLB2T - C2)))
dORI/dt= - kdori*ORI + (ksori*((CLB5T-C5)+eorib2*(CLB2T - C2)))
dC2/dt= - ((kdib2+Vdb2+kd1c1+(Vd2c1/(Jd2c1+SIC1T))))*C2 +
          (kasb2*(CLB2T - C2)*(SIC1T -(C2+C5)))
dSPN/dt= - kdspn*SPN + (ksspn*((CLB2T - C2)/(Jspn+(CLB2T - C2))))
dCLN2/dt= - kdn2*CLN2 + ((ksn2'+ksn2''*SBF)*MASS)
dMASS/dt=mu*MASS
dVi20/dt=0
dBeta/dt=0
dTimer1/dt=1

```

```

dTimer2/dt=(1/12)
dAlpha/dt=0
#Globals
global -1 {(CLB2T-C2) -0.3 } {MASS = f*MASS; SPN=0.0000; BUD=0.0000 }
global -1 {(CLB2T-C2+CLB5T-C5)-0.2 } {ORI=0.0000 }
global 1 {ORI-1 } {Vi20 =10; Alpha=1 }
global 1 {SPN-1 } {Alpha=0.0000000000; Beta=1; Timer1=0; Timer2=0 }
global 1 {Timer2-1 } {Beta=0.0000000000; Vi20=0.100; Alpha=1 }
#Initial Conditions
init CLB2T=.007, SIC1T=.97, C5=.03, CDC20=.0625
init HCT1=1, BUD=.065, CLB5T=.06, CDC20T=.2420
init ORI=.015625, C2=.003, SPN=.001, CLN2=.02
init MASS=.97, Vi20=0.100, Beta=0, Timer1=0
init Timer2=2, Alpha=1
#Constants
param Jd2c1=0.05, HCT1T=1, kisbf''=6, Jisbf=0.01
param f=.433, kd20=0.08, Jasbf=0.01, kd1c1=0.01
param kisbf'=0.5, BCK20=.0027, kaswi=1, kiswi'=0.3
param kiswi''=0.2, Jaswi=.1, Jiswi=.1, kssp=0.08
param Jspn=0.2, kdspn=0.06, mu=0.005776, kdori=0.06
param ksbud=0.3, ebudb5=1, kdbud=0.06, kamcm=1
param kimcm=0.15, Jamcm=1, Jimcm=1, kd2c1=.3
param ec1n3=20, ec1k2=2, ec1b5=1, ec1b2=0.067
param ksori=2, eorib2=0.4, kasbf=1, esbfn3=75
param esbfb5=.5, kat1'=.04, kat1''=2, Jat1=.05
param Jit1=.05, kit1'=0, kit1''=.64, eit1n2=1
param eit1b5=.5, eit1b2=1, ka20=1, kdib5=0.05
param ks20'=0.005, ks20''=0.06, kasb2=50, kdib2=0.05

```

```
param kasb5=50, ksn2'=0, ksn2''=.05, kdn2=0.1
param ksb2'=0.002, ksb2''=0.05, kdb2'=0.01, kdb2''=2
param kdb2'''=.05, ksb5'=0.006, ksb5''=0.02, kdb5'=0.1
param kdb5'''=0.25, CLN3MAX=0.02, Dn3=1, Jn3=6
param ksc1'=0.02, ksc1''=0.1
#Plot dependent species
done
*****OUTPUT FILE (budding yeast)*****
```

Bibliography

1. Alur, Rajeev; Belta, Calin; Kumar, Vijay; Mintz, Max; Pappas, George J.; Rubin, Harvey; and Jonathan Schug. "Modeling and Analyzing Biomolecular Networks," *Computing in Science and Engineering*, 2–13 (January 2002).
2. Alur, Rajeev; Hur, Yeran; Ivančić, Franjo; Lee, Insup; and Oleg Sokol-sky. *CHARON User Manual (Version 0.9)*. University of Pennsylvania, 2002. <http://www.cis.upenn.edu/mobies/charon/manual.ps>.
3. Arkin, Adam; Ross, John; and Harley H. McAdams. "Stochastic Kinetic Analysis of Developmental Pathway Bifurcation in Phage λ -Infected Escherichia coli Cells," *Genetics Society of America*, 149:1633–1648 (August 1998). http://insilico.mit.edu/Genetics_149_1633.pdf.
4. Belta, Calin; Schug, Jonathan; Dang, Thao; Kumar, Vijay; Pappas, George J.; Rubin, Harvey; and Paul Dunlap, "Stability and Reachability Analysis of a Hybrid Model of Luminescence in the Marine Bacterium *Vibrio Fischeri*." Internet. http://www.cis.upenn.edu/biocomp/CDC_REG1559.pdf.
5. BioSPICE, "Modeling and Simulation of Bioregulatory Networks in BIOCHARON." Internet (BioSPICE Community server: key documents—models and tools). <https://community.biospice.org/>.
6. Black, Jacquelyn G. *Microbiology Principles and Explorations* (Fourth Edition). NJ: Prentice Hall, 1999.
7. Brauer, Fred and John A. Nohel. *Qualitative Theory of Ordinary Differential Equations*. NY: Dover Publications Inc., 1969.
8. Chen, Katherine C.; Csikasz-Nagy, Attila; Gyorffy, Bela; Val, John; Novak, Bela; and John J. Tyson. "Kinetic Analysis of a Molecular Model of the Budding Yeast Cell Cycle," *Molecular Biology of the Cell*, 11:369–391 (January 2000).
9. Fogel, Gary B. and David W. Corne, editors. *Evolutionary Computation in Bioinformatics*. Amsterdam: Morgan Kaufmann Publishers, 2003.
10. Hasty, Jeff; McMillen, David; Isaacs, Farren; and James J. Collins. "Computational Studies of Gene Regulatory Networks: In Numero Molecular Biology," *Nature Reviews Genetics*, 2:268–279 (April 2001). www.nature.com/reviews/genetics.
11. Hasty, Jeff; Pradines, Joel; Dolnik, Milos; and J. J. Collins. "Noise-based switches and amplifiers for gene expression," *Proceedings of the National Academy of Sciences of the United States of America*, 97(5):2075–2080 (February 2000). <http://web.mit.edu//biophysics/qbio/PDFs/Hasty.pdf>.

12. Heinrich, R. and S. M. Rapoport. "Metabolic Regulation and Mathematical Models," *Progress in Biophysics and Molecular Biology*, 32:1–82 (1977).
13. Human Genome Program. *Primer on Molecular Genetics*. U.S. Department of Energy, Office of Energy Research, and Office of Health and Environmental Research, 1992. <http://www.esp.org/misc/genome/primer.pdf>.
14. Karp, Richard M. "Mathematical Challenges from Genomics and Molecular Biology," *Notices Of The AMS*, 49(5):544–553 (May 2002).
15. Keio University and Mitsui Knowledge Industry Company, Ltd. *E-CELL2 User's Manual*. Keio University, 2002. <http://www.bioinformatics.org/naota/ecell2/manual220/indexE.html>.
16. MathWorks, "The Benefits of MATLAB." Internet, 2003. <http://www.mathworks.com/products/matlab/description1.jsp>.
17. Mendes, Pedro, "Gepasi Biochemical Simulation." Internet, December 2002. <http://www.gepasi.org/>.
18. Morikawa, Yoshitomi; Yamaguchi, Tomohiko; and Takashi Amemiya. "Analytical Approach for Oscillation Properties of Soft Materials," *Forma*, 15:249–256 (2000). <http://www.scipress.org/journals/forma/pdf/1503/15030249.pdf>.
19. Neufeldt, Victoria and David B. Guralnik, editors. *Webster's New World Dictionary of American English—Third College Edition*. OH: MacMillan, Inc., 1994.
20. Novak, Bela, "Mathematical Modeling of the Budding Yeast Cell Cycle." Talk at the HHMI Meeting of the International Research Scholars, HHMI, Chevy Chase, June 2000. <http://www.celcycle.bme.hu/bnovak/pdfek/chevy%20chase/talk.pdf>.
21. Nuwaysir, Emile F.; Bittner, Michael; Trent, Jeffrey; Barrett, J. Carl; and Cynthia A. Afshari. "Microarrays and Toxicology: The Advent of Toxicogenomics," *Molecular Carcinogenesis*, 24:153–159 (1999).
22. Prigogine, I. and R. Lefever. "Symmetry Breaking Instabilities in Dissipative Systems II," *The Journal of Chemical Physics*, 48(4):1965–1700 (February 1968).
23. Schaff, J. and F. M. Loew, "The Virtual Cell." Pacific Symposium on Biocomputing, 1999. <http://www.nrcam.uchc.edu/publications/>.
24. Schnackenberg, J. "Simple Chemical Reaction Systems with Limit Cycle Behavior," *Journal of Theoretical Biology*, 81:389–400 (1979).
25. Skinner, Gordon B. *Introduction To Chemical Kinetics*. NY: Academic Press, Inc., 1974.
26. Smolen, Paul; Baxter, Douglas A.; and John H. Byrne. "Modeling Transcriptional Control in Gene Networks—Methods, Recent Results,

- and Future Directions,” *Bulletin of Mathematical Biology*, 247–292 (2000).
<http://www.idealibrary.com>.
27. Templin, Jay M. *The Essentials of Biochemistry*. Research and Education Association, Piscataway NJ, 1998.
 28. Tomita, Masaru, “E-CELL: Towards Integrative Modeling of Cellular Processes.” Internet, December 2002.
<http://incob.biotec.or.th/pdf/MasaruTomita.pdf>.
 29. U.S. Department of Energy Human Genome Program, “Genomics and Its Impact on Medicine and Society, A 2001 Primer.” Internet, 2001.
<http://www.ornl.gov/hgmis/publicat/primer2001/primer11.pdf>.
 30. Vass, Marc; Shaffer, Clifford A.; Tyson, John J; Ramakrishnan, Naren; and Layne T. Watson, “The JigCell Model Builder: A Tool for Modeling Intra-Cellular Regulatory Networks.” Internet, December 2002.
<http://gnida.cs.vt.edu/cellcyclepse/modelbuilder.pdf>.
 31. Virginia Tech, “The Eukaryotic Cell Cycle Collaborative Problem-Solving Environment Group.” Internet, March 2003.

Vita

2LT Young was born in Dover, Delaware. He also grew up in Dover where he graduated from a nearby high school in 1986. He enlisted in the United States Air Force in August 1986.

As an enlisted Air Force member, he served as a cable and antenna installations/maintenance specialist. He was assigned to Griffiss Air Force Base (AFB), New York; Yokota Air Base, Japan; McClellan AFB, California; Howard AFB, Panama. In September 1999, he separated from active duty Air Force under the *Professional Officer's Course Early Release Program* and entered into the Reserve Officers Training Corps to complete his undergraduate degree at Charleston Southern University, South Carolina. He graduated with a Bachelor of Science in Mathematics/Computer Science in July 1999.

2LT Young's first assignment as a commissioned officer was to attend the Air Force Institute of Technology (AFIT) in residence at Wright-Patterson AFB, Ohio to complete a Master of Science in Computer Science.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188		
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 03-17-2003		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Mar 2002 - Mar 2003	
4. TITLE AND SUBTITLE DETERMINISTIC INTRACELLULAR MODELING			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Young, Jacqueline, B., Second Lieutenant, USAF			5d. PROJECT NUMBER ENR #2003-017		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GCS/ENC/03M-1		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR Attn: Dr. Kozumbo 801 N. Randolph St. Room 732 Arlington, VA 22203 Phone: (703) 696-7720			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>The United States Air Force is interested in the potential side effects—at the cellular level—from exposure to mission-essential chemicals. Presently, Air Force toxicology studies are conducted to help shed light in identifying potential hazards to workers. However, it takes a considerable amount of money, resources, and time to obtain and analyze experimental results from toxicology studies. The necessity for innovative methods that enable researchers to more effectively generate and analyze data is apparent.</p> <p>Mathematical modeling is a viable option to become a valuable tool for the researcher. Mathematical models can rapidly generate informative predictions on how a cell reacts to a certain toxicant exposure. Moreover, information is readily available when generated by mathematical models.</p> <p>This research involves the study of one non-biological reaction system and four biological, intracellular reaction systems. Each system is converted into a mathematical model using the rate-equation approach. Numerical simulation results from these mathematical models are obtained using two novel software modeling tools and MATLAB. Results obtained from the novel modeling tools are compared to MATLAB's results in order to ascertain the accuracy of each novel modeling tool.</p> <p>The experience that is gained in deriving mathematical models and using novel tools to perform numerical simulations for these reaction systems should help the Air Force develop intracellular models to assist in future toxicology studies.</p>					
15. SUBJECT TERMS Cellular Biology, Biochemistry, Biological Pathways, Protein Synthesis, Mathematical Modeling, BioSpice, Intracellular Modeling					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dennis W. Quinn, ENC
a. REPORT	b. ABSTRACT	c. THIS PAGE			
U	U	U	UU	131	19b. TELEPHONE NUMBER (Include area code) (937) 255-3636, ext 4522; e-mail: Dennis.Quinn@afit.edu