Mass Fluctuation Kinetics: Analysis and Computation of Equilibria and Local Dynamics

by

Paul Azunre

B.A., Economics (2007)
B.S., Engineering (2007)
Swarthmore College

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2009

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
January 29, 2008

Certified by . . . .
George Verghese
Professor of Electrical Engineering
Thesis Supervisor

Accepted by . . . .
Terry P. Orlando
Chair, Department Committee on Graduate Students

Mass Fluctuation Kinetics: Analysis and Computation of Equilibria and Local Dynamics

by

Paul Azunre

Submitted to the Department of Electrical Engineering and Computer Science
on January 29, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

The *mass fluctuation kinetics* (MFK) model is a set of coupled first-order differential equations describing the temporal evolution of means, variances and covariances of species concentrations in systems of chemical reactions. It generalizes classical *mass action kinetics* (MAK) in which fluctuations around the mean are ignored. This thesis begins with the motivating background theory for the development of MFK. The model equations follow from the time-evolution of the molecule number *moment generating function* obtained from the *chemical master equation* (CME). A closed-form expression for the MFK *Jacobian matrix* that describes small deviations from equilibrium is derived. An MFK software toolbox prototype, developed in MATLAB (and available at `http://www.mit.edu/~azunre/MFK`), applies this Jacobian in the context of *single substrate enzyme kinetics* to exploring the local dynamics of MFK equilibria. MFK means and covariances are observed to be locally decoupled at the equilibrium in the large-volume *thermodynamic limit*, providing an alternative explanation for why MAK is an accurate approximation for system behavior there. Increasing discreteness of system behavior with decreasing system volume, a characteristic that the MAK model cannot capture, is captured by the MFK model via the growth of its variance. This ability is limited to a threshold beyond which MFK ceases to be a useful approximation for system behavior. Systematic extensions to higher order moments to correct for this are suggested.

Thesis Supervisor: George Verghese
Title: Professor of Electrical Engineering

Dada, this one is for you

Acknowledgements

I hereby find it necessary to express my gratitude, in no particular order, to various people whose direct and indirect contributions made this work possible.

To my research supervisor, George Verghese, for the opportunity to carry out this work in spite of my deficiency of related experience, for funding it over a summer month, for the countless suggestions and perhaps most importantly, for the invariable patience and encouragement. To Carlos Gomez-Uribe, who together with George conceived MFK, for many invaluable suggestions at various stages of this project. To my academic counsellor, Gregory Wornell, for keeping me on track. To Ricardo Paxson, my summer research supervisor at The MathWorks, Inc., who oversaw a rapid growth in confidence as a researcher that spilled significantly into this project. To Carr Everbach, my undergraduate research supervisor at Swarthmore College, who oversaw my birth as a researcher and guided me through the roughest patches of my undergraduate experience. To Marc Baldo, for catching me on a downward spiral and reminding me to keep the faith. To the MIT Department of Electrical Engineering and Computer Science, for the three semesters' worth of Teaching Assistantships that funded my Master's degree and financially supported this work. To my friends at MIT, Demba Ba, Brian Omwenga and Anima Singh, for the conversations, academic and otherwise, that helped me keep my grip on sanity.

To my brother Richard, a partner in our nomadic explorations of this planet, for irreplaceable company in dealing with the associated challenges. To Tane, with whom I have had the pleasure to spend the past two years of my life, for keeping me human. To my junior siblings, Gideon and Gifty, whose precious company I have had to forego for some time now, but who are always with me. To my grandmothers, two special women who shaped my personality in very unique ways. To my mother, who faced great adversity and made uncountable sacrifices to raise me to be me. To my late father, who even in his absence influences what I do in a way that no other person can.

Contents

1 Stochastic Chemical Kinetics **15**

1.1 Introduction 15

1.2 System of chemical reactions 16

1.3 Molecular dynamics 18

1.4 System Markov process model 18

1.5 The stochastic simulation algorithm (SSA) 20

1.6 The tau-leaping algorithm 21

1.7 The chemical Langevin equation (CLE) 22

1.8 Mass action kinetics (MAK) 23

1.9 Summary 24

2 Mass Fluctuation Kinetics (MFK) **25**

2.1 Introduction 25

2.2 Preliminaries 26

2.3 Moment generating function evolution 27

2.4 Mean evolution 28

2.5 Covariance evolution 30

2.6 Summary and future work 33

3 MFK Jacobian Matrix **35**

3.1 Introduction 35

3.2 The vectorized model 36

3.3 The Jacobian matrix 37

4 MFK Software Prototype **39**

4.1 Introduction 39

4.2 Technical specifications 39

4.3 Capabilities and supporting theory 40

4.3.1 Preparing and displaying the MFK description 40

4.3.2 Plotting trajectories 43

4.3.3 Finding equilibria via Newton-Raphson 44

4.3.4 Local bifurcation analysis 45

4.3.5 Local mean-covariance coupling analysis 47

5 MFK Enzyme Kinetics **49**

5.1 Introduction 49

5.2 Single substrate enzyme reaction system 50

5.3 Analysis and discussion 50

5.3.1 Complex formation and dissociation (CFD) 50

5.3.2 Irreversible enzyme reaction system (IERS) 59

5.3.3 Reversible enzyme reaction system (RERS) 67

5.4 Conclusions and future work 73

A Matrix Theory Overview **75**

A.1 Matrices and vectors 75

A.2 Vectorization 77

A.3 The Kronecker product 78

A.4 The Khatri-Rao product 79

A.5 Matrix derivatives 79

List of Figures

4-1 Two iterations of the Newton Raphon algorithm on a simple example . . 45

5-1 [Color] PrepareModel function call and output for CFD 55

5-2 [Color] BifurcationPlot function call and text output for CFD 58

5-3 [Color] Local bifurcation plot for CFD 60

5-4 [Color] CFD trajectories in stable MFK regime 60

5-5 [Color] CFD trajectories in (approximately) marginally stable MFK regime 61

5-6 [Color] CFD trajectories in unstable MFK regime 61

5-7 [Color] Participation factor evolution for CFD 62

5-8 [Color] PrepareModel function call and output for IERS 64

5-9 [Color] BifurcationPlot function call and text output for IERS 65

5-10 [Color] Local bifurcation plot for IERS 66

5-11 [Color] IERS trajectories in stable MFK regime 67

5-12 [Color] IERS trajectories in (approximately) marginally stable MFK regime 68

5-13 [Color] IERS trajectories in unstable MFK regime 68

5-14 [Color] Test of IERS SSA molecule number assertion 69

5-15 [Color] Participation factor plot for IERS 69

5-16 [Color] Local bifurcation plot for RERS 71

5-17 [Color] RERS trajectories in stable MFK regime 71

5-18 [Color] RERS trajectories in (approximately) marginally stable MFK regime 72

5-19 [Color] RERS trajectories in unstable MFK regime 72

5-20 [Color] Participation factor plot for RERS 73

Preface

The high-level purpose of this text is to provide a self-contained exposition of the *mass fluctuation kinetics* (MFK) model for the temporal evolution of means, variances and covariances of concentrations of species in systems of chemical reactions. The motivating background theory for the model's development is provided in Chapter 1. The presentation of concepts here is heavily influenced by Daniel Gillespie's lecture on stochastic chemical kinetics, a video of which is available at `http://mbi.osu.edu/2004/ws2abstracts.html`. The remaining chapters contain several original contributions to the MFK framework.

The model equations are shown in Chapter 2 to follow from the temporal evolution of the molecule number *moment generating function* obtained from the *chemical master equation* (CME). This connection highlights the possibility of systematically extending the MFK model to specific higher-order moments and studying their effects on system behavior. A general analytic result for computing evolution equations up to any order has (independently of our work) been established in [1] via a similar approach.

An expression for the MFK *Jacobian matrix* is established in Chapter 3.

Chapter 4 describes a software toolbox prototype developed in MATLAB, a popular numerical computing environment, to automate the analysis of systems within the MFK framework. The toolbox is available for download at the companion website for this thesis, `http://www.mit.edu/~azunre/MFK`. Chapter 5 applies the toolbox to exploring the local dynamics of MFK equilibria, and relating them to the local dynamics of the traditional *mass action kinetics* (MAK) model in the context of *single substrate enzyme kinetics*. This exploration is backed by analytic justification as much as possible, to enable it serve as a tutorial for anyone interested in studying the model and/or applying the software toolbox. *Bifurcation movies* described in Chapter 5 and an electronic (.pdf) version of this text are available for view and download at the companion website. Appendix A describes key matrix results used in arriving at the equations in Chapter 3 (and to a smaller extent, Chapter 2). Several relevant matrix identities not covered by

Appendix A can be found at `http://matrixcookbook.com`.

Chapter 1

Stochastic Chemical Kinetics

1.1 Introduction

Evolution with time of concentrations of reacting species in systems of chemical reactions is traditionally modeled in chemical kinetics by deterministic, continuous reaction rate equations based on *mass action kinetics* (MAK). These equations dictate that the rate of a chemical reaction is proportional to the product of the concentrations of reacting species, with constant of proportionality known as the reaction's *rate constant.* While this description is empirically accurate for systems of test-tube size or larger, it breaks down at smaller volumes found in, for example, gene networks and cellular signaling pathways, where stochastic effects can dominate and where the discrete nature of the evolution of molecule numbers becomes significant. Stochastic chemical kinetics is concerned with developing modeling methodologies that correct for this deficiency by introducing elements of stochasticity, and perhaps discreteness, into the rate equations.

This chapter reviews popular approaches and their theoretical connection to the deterministic MAK reaction rate equations. The purpose of this chapter is to provide the background and motivation for the development of the *mass fluctuation kinetics* (MFK) model. The next section describes the theoretical setting for an arbitrary system of chemical reactions.

1.2 System of chemical reactions

Consider molecules of n chemical species, $\{X_i\}_{i=1}^{n}$, known as the *reactants*, interacting through reactions of the form

$$\sum_{i=1}^{n} \underline{s}_i X_i \xrightarrow{k} \sum_{i=1}^{n} \overline{s}_i X_i \tag{1.1}$$

in a constant volume v. Each such reaction may be broken down into *elemental* reactions that describe *single instantaneous physical events*, i.e., either collisions of sufficiently high energy of two or more reactant molecules, or spontaneous molecule formation/dissociation events. Since instantaneous collisions of three or more reactant molecules are extremely rare, elemental reactions may be restricted, without loss of generality, to the four categories in Table 1.1. Note that $\varnothing$ denotes the empty set.

	Reaction category	**Representation**
1.	*zero-order* reaction	$\varnothing \xrightarrow{k} X_i$
2.	*first-order* reaction	$X_j \xrightarrow{k} \sum_{i=1}^{n} \overline{s}_i X_i$
3.	*homogeneous second-order* reaction	$X_j + X_j \xrightarrow{k} \sum_{i=1}^{n} \overline{s}_i X_i$
4.	*heterogeneous second-order* reaction	$X_j + X_l \xrightarrow{k} \sum_{i=1}^{n} \overline{s}_i X_i \quad j \neq l$

Table 1.1: *Elemental reactions and their representations.*

It is assumed that the system is completely described by L elemental reactions, depicted as

$$\left\{ R_l : \sum_{i=1}^{n} \underline{s}_{li} X_i \xrightarrow{k_l} \sum_{i=1}^{n} \overline{s}_{li} X_i \right\}_{l=1}^{L} \tag{1.2}$$

and the qualifier elemental is henceforth dropped. Parameter k_l is the *rate constant* for reaction R_l, the constant of proportionality for its MAK reaction rate equation. Upon each firing of R_l, $\underline{s}_{li}$ molecules are consumed and $\overline{s}_{li}$ molecules of X_i are produced. The *stoichiometric coefficient* of X_i in R_l, defined as the change in the number of X_i molecules upon each firing of R_l, is calculated as $s_{li} = \overline{s}_{li} - \underline{s}_{li}$. Stoichiometric coefficients are grouped into the *stoichiometry vector* for each R_l as

$$s_l = \begin{bmatrix} s_{l1} \\ \vdots \\ s_{ln} \end{bmatrix} \in \mathbb{Z}^n \tag{1.3}$$

Stoichiometry vectors are arranged into the system's *stoichiometry matrix* as

$$\mathbf{S} = \begin{bmatrix} s_1 & \cdots & s_L \end{bmatrix} \in \mathbb{Z}^{n \times L} \tag{1.4}$$

System size is defined as

$$\Omega = Av \tag{1.5}$$

where A is *Avogadro's number*, the number of molecules in one *mole* of a chemical species.

Molecule numbers for all species at time t may be grouped as the column vector

$$x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix} \in \mathbb{Z}^n \tag{1.6}$$

and concentrations may be obtained in units of *moles per unit volume* as

$$y(t) = \frac{x(t)}{\Omega} \in \mathbb{R}^n \tag{1.7}$$

The system is assumed to be evolving at a constant temperature T, hence to have achieved *thermal equilibrium*, but not *chemical equilibrium*. Temporal evolution of the system towards this chemical equilibrium (if it exists and is attained, otherwise the nature of its unstable behavior) is of interest. The next section describes *molecular dynamics*, a detailed method for characterizing this evolution.

1.3 Molecular dynamics

A detailed characterization of the system's temporal evolution may be obtained via molecular dynamics [2]. This method involves specifying initial positions and velocities of every molecule in the system and tracking their evolution with time based on the deterministic laws of classical mechanics. It follows from physical principles that when two molecules collide with sufficiently high energy (specifically, energy greater than the reaction's *activation energy*), a second-order reaction occurs and appropriate molecule numbers in $x(t)$ are updated. This describes a reactive collision. All other collisions are nonreactive in that they merely alter the motions of colliding molecules. Zero-order and first-order reactions are driven by underlying molecular processes, which introduce an element of stochasticity into the evolution of $x(t)$ if reactions of this type exist in the system.

This method tracks an enormous amount of state information, making it computationally intensive and infeasibly slow for most practical systems. The assumption of *spatial homogeneity* allows a significant truncation of state information at the expense of treating $x(t)$ as a *stochastic process* regardless of reaction categories in the system. The next section makes this assumption and establishes a continuous time, discrete state *Markov process* model for $x(t)$.

1.4 System Markov process model

If reactive collisions are assumed to be separated in time by many nonreactive collisions, the amount of state information may be significantly reduced. Under this assumption, the system is kept well-stirred, and hence spatially homogeneous, by the nonreactive collisions, thus allowing the reactive collisions and internal molecular processes to occur at uniform rates throughout the system. Thus, nonreactive collisions may be ignored and attention restricted to the zero-, first- and second-order reactions. The benefit of this state information truncation is that $x(t)$ can be reasonably modeled as a Markov process with well-defined *exponential state transition rates* that need to be specified. The treatment below draws from the work of Daniel Gillespie, particularly from his lecture on stochastic chemical kinetics available at `http://mbi.osu.edu/2004/ws2abstracts.html`.

The *propensity function* $a_l(\mathbf{x})$ for each reaction R_l is defined such that the probability of R_l firing within the next infinitesimal time interval dt is given by $a_l(\mathbf{x})dt$ (in the absence of all other reactions and given that the current state is $\mathbf{x}$). Under the standing assumptions, $a_l(\mathbf{x})$ may be derived from physical principles for each R_l depending on its category, as shown in Table 1.2, with c_l for a fixed Ω denoting a constant associated with R_l. Note that $x_i = \{\mathbf{x}\}_i$ is the i^{th} entry of $\mathbf{x}$.

	Reaction category	**Propensity function** $[a_l(\mathbf{x})]$
1.	zero-order reaction	c_l
2.	first-order reaction	$c_l x_i$
3.	homogeneous second-order reaction	$c_l \frac{x_i(x_i-1)}{2}$
4.	heterogeneous second-order reaction	$c_l x_i x_j \qquad i \neq j$

Table 1.2: *Propensity functions expressed in terms of physical rate constants.*

For zero-order and first-order reactions, the constant parameter c_l arises from the rates of the underlying spontaneous formation/dissociation events. For zero-order reactions, it specifies the probability that spontaneous formation will occur anywhere in the system (or can represent inflow of species). For first-order reactions, it specifies the probability that a randomly chosen molecule of the reactant will undergo spontaneous dissociation, while the term x_i reflects the number of such molecules that may be randomly chosen. For second-order reactions, the constant parameter c_l arises from collision theory arguments and specifies the probability that a randomly chosen reactant molecule pair will collide with sufficiently high energy; in this second-order case, c_l varies inversely with Ω. The terms $\frac{x_i(x_i-1)}{2}$ and $x_i x_j$ reflect the number of such distinct molecule pairs, for homogeneous and heterogeneous reactions respectively, that may be randomly chosen. Since every c_l arises from the physics describing the system, they are collectively referred to as the *physical rate constants.*

A continuous time, discrete state Markov process model for the system of chemical reactions, with state $x(t) \in \mathbb{Z}^n$ and *exponential state transition rates* $\{a_l(\mathbf{x})\}_{l=1}^{L}$, has thus been established. Transitions out of an arbitrary state $\mathbf{x}$ into the states $\{\mathbf{x}+s_l\}_{l=1}^{L}$ correspond to the firing of reactions $\{R_l\}_{l=1}^{L}$ at rates $\{a_l(\mathbf{x})\}_{l=1}^{L}$, respectively. The total transition rate out of $\mathbf{x}$ is then $\sum\limits_{l=1}^{L} a_l(\mathbf{x})$. Similarly, transitions out of states $\{\mathbf{x}-s_l\}_{l=1}^{L}$ into the state $\mathbf{x}$ occur at rates $\{a_l(\mathbf{x}-s_l)\}_{l=1}^{L}$ respectively. The *forward Kolmogorov equation* for the Markov process may then be written as

$$\frac{dP_{\mathbf{x}}(t)}{dt} = \sum_{l=1}^{L} \left[P_{\mathbf{x}-s_l}(t)a_l(\mathbf{x}-s_l) - P_{\mathbf{x}}(t)a_l(\mathbf{x})\right] \tag{1.8}$$

to describe the rate of change of $P_{\mathbf{x}}(t)$, the probability of being in state $\mathbf{x}$ at time t conditioned on some initial state $x(0)$. This is the *chemical master equation* (CME), which completely characterizes the temporal evolution of the system's stochasticity under the standing assumptions.

Since the number of possible states is typically large, solving the CME explicitly is infeasible in most practical situations. The *stochastic simulation algorithm* (SSA), which computes sample trajectories from the CME without solving the full set of equations for each possible state, alleviates this drawback and is presented in the next section.

1.5 The stochastic simulation algorithm (SSA)

The stochastic simulation algorithm (SSA) [3] computes sample trajectories from the CME without solving it for each possible state. Let $p(j,\tau|\mathbf{x})$ be the joint probability density function between the index of the next reaction to fire, $j \in \{1, ..., L\}$, and the time until it fires, τ, given that $x(t) = \mathbf{x}$ at the current time t. Then

$$\begin{aligned} p(j,\tau|\mathbf{x})d\tau &= \text{Prob. no reaction in } [t, t+\tau) \text{ and } j^{th} \text{ reaction in } [t+\tau, t+\tau+d\tau) \\ &= e^{-\left(\sum\limits_{l=1}^{L} a_l(\mathbf{x})\right)\tau} \cdot a_j(\mathbf{x})d\tau = \overbrace{\left(\sum_{l=1}^{L} a_l(\mathbf{x})\right) e^{-\left(\sum\limits_{l=1}^{L} a_l(\mathbf{x})\right)\tau}}^{p_1(\tau)} \cdot \overbrace{\frac{a_j(\mathbf{x})}{\left(\sum\limits_{l=1}^{L} a_l(\mathbf{x})\right)}}^{p_2(j)} d\tau \end{aligned} \tag{1.9}$$

It should be clear from Equation (1.9) that a random sample (j, τ) may be drawn from $p(j, \tau|\mathbf{x})$ by drawing independent samples from the exponential $p_1(\tau)$ and the uniform $p_2(j)$. This is done at every step of the algorithm, with molecule numbers updated as $x(t + \tau) = \mathbf{x} + s_j$ and time advanced by τ.

Since samples are drawn (i.e., reactions are simulated) one at a time, this method can be prohibitively slow (although not infeasible with sufficient computing power) in practical situations where many molecules are present and reactions fire frequently. The *tau-leaping* algorithm, which approximates the SSA, alleviates this drawback and is presented in the next section. Yet other approaches to efficient stochastic simulation may be found in the literature, see for example [4].

1.6 The tau-leaping algorithm

Typically, a time interval τ (over which each reaction may fire several times) can be adaptively found such that the expected changes induced in the propensity functions $\{a_l(\mathbf{x})\}_{l=1}^{L}$ are relatively small. This is known as the *leap condition* and reliable procedures have been developed for finding suitable candidates [5]. If such a τ can be found, the propensity functions $\{a_l(\mathbf{x})\}_{l=1}^{L}$ may be assumed to remain constant over τ. Then, the number of times that each R_l fires over τ is a sum of a random number of independent exponential random variables of rate $a_l(\mathbf{x})\tau$, and hence is a Poisson random variable with mean and variance equal to $a_l(\mathbf{x})\tau$. Let this Poisson random variable be denoted by $\mathcal{P}(a_l(\mathbf{x})\tau)$.

Reliable procedures exist for generating random samples from Poisson probability distributions. Thus, the SSA may be approximated by drawing L Poisson random samples, $\{\mathcal{P}(a_l(\mathbf{x})\tau)\}_{l=1}^{L}$, updating molecule numbers as

$$x(t + \tau) = \mathbf{x} + \sum_{l=1}^{L} s_l \mathcal{P}(a_l(\mathbf{x})\tau) \tag{1.10}$$

and advancing time by τ. This is the tau-leaping algorithm [6].

The *chemical Langevin equation* (CLE) may be viewed as a direct consequence of tau-leaping and provides a bridge to the MAK reaction rate equations. It is presented in the next section.

1.7 The chemical Langevin equation (CLE)

If a τ can be found to satisfy the leap condition such that the quantities $\{a_l(\mathbf{x})\tau\}_{l=1}^{L}$ are sufficiently larger than 1, then the Poisson random variables $\{\mathcal{P}(a_l(\mathbf{x})\tau)\}_{l=1}^{L}$ may be approximated by normal random variables with means and variances both equal to $\{a_l(\mathbf{x})\tau\}_{l=1}^{L}$. Let these be denoted as $\{\mathcal{N}_l(a_l(\mathbf{x})\tau, a_l(\mathbf{x})\tau)\}_{l=1}^{L}$.

If such a τ can be found, the tau-leaping state update equation (1.10) may be approximated as

$$\begin{aligned} x(t+\tau) &= \mathbf{x} + \sum_{l=1}^{L} s_l \mathcal{N}_l(a_l(\mathbf{x})\tau, a_l(\mathbf{x})\tau) \\ &= \mathbf{x} + \sum_{l=1}^{L} s_l a_l(\mathbf{x})\tau + \sum_{l=1}^{L} s_l \sqrt{a_l(\mathbf{x})\tau}\mathcal{N}_l(0,1) \end{aligned} \tag{1.11}$$

where $\mathcal{N}_l(0,1)$ is a zero-mean unit-variance normal random variable. It can be shown that this equation reduces to

$$\frac{dx(t)}{dt} = \sum_{l=1}^{L} s_l a_l(x(t)) + \sum_{l=1}^{L} s_l \sqrt{a_l(x(t))}\Gamma_l(t) \tag{1.12}$$

where $\{\Gamma_l(t)\}_{l=1}^{L}$ are independent unit-intensity zero-mean *Gaussian white noise* processes. This is the chemical Langevin equation (CLE) [7]. Technically, it is a *stochastic differential equation* (SDE). It should be noted that in moving from Poisson random variables to normal random variables, discrete system temporal evolution is being approximated as continuous. The *thermodynamic limit* of the CLE yields the MAK reaction rate equations, as shown in the next section.

1.8 Mass action kinetics (MAK)

The thermodynamic limit of the system is defined as its limiting behavior with large molecule numbers, $x(t)$, and system size, Ω, i.e.,

$$\text{thermodynamic limit } \overset{\triangle}{=} \lim_{\{x(t),\Omega\}\to\infty}\{\text{system}\} \tag{1.13}$$

such that concentrations, $y(t) = \frac{x(t)}{\Omega}$, remain unchanged. It may be shown that substituting $x(t) = y(t)\Omega$ in (1.12) and taking the limit of $\Omega \to \infty$ yields the traditional set of MAK reaction rate equations in terms of concentrations rather than molecule numbers:

$$\frac{dy(t)}{dt} = \lim_{\Omega\to\infty}\left\{\sum_{l=1}^{L} s_l \frac{a_l(y(t)\Omega)}{\Omega}\right\} = \sum_{l=1}^{L} s_l \bar{r}_l(y(t)) \tag{1.14}$$

Microscopic reaction rates are defined as $\left\{\rho_l(y(t)) \overset{\triangle}{=} \frac{a_l(y(t)\Omega)}{\Omega}\right\}_{l=1}^{L}$ for reactions $\{R_l\}_{l=1}^{L}$ respectively. Taking the limit of $\Omega \to \infty$ yields the *macroscopic* reaction rates

$$\left\{\bar{r}_l(y(t)) \overset{\triangle}{=} \lim_{\Omega\to\infty} \rho_l(y(t))\right\}_{l=1}^{L} \tag{1.15}$$

We list in Table 1.3 the microscopic and macroscopic reaction rates and make explicit the definition of the rate constant k_l in terms of the corresponding c_l for each reaction category. Note that $y_i = \{y(t)\}_i$ is the i^{th} entry of $y(t)$.

	Reaction category	$\rho_l(y(t))$	$\bar{r}_l(y(t))$	k_l
1.	zero-order reaction	c_l	k_l	c_l
2.	first-order reaction	$c_l y_i$	$k_l y_i$	c_l
3.	homogeneous second-order reaction	$\frac{c_l}{2} y_i\,(y_i\Omega - 1)$	$k_l y_i^2$	$\lim_{\Omega\to\infty} \frac{c_l\Omega}{2}$
4.	heterogeneous second-order reaction	$c_l\Omega y_i y_j \quad i \neq j$	$k_l y_i y_j \quad i \neq j$	$\lim_{\Omega\to\infty} c_l\Omega$

Table 1.3: *Reaction rates and relationships between rate constants.*

1.9 Summary

The MAK reaction rate equations have been shown to be the thermodynamic limit of a more detailed and accurate model of system evolution. In this limit, the stochasticity of the system becomes negligible in comparison to the deterministic part and drops out of the reaction rate equations. Applications far from the thermodynamic limit may be of interest. The CME, which is a detailed characterization of the system's stochasticity evolution under the assumption of spatial homogeneity, is infeasible to solve in most practical situations. The SSA and the tau-leaping algorithm can compute sample trajectories from the CME, but do not provide a complete picture of system evolution. *Monte Carlo* methods can be used to approximate moments of the CME but are significantly more computationally intensive.

The remainder of this text focuses on mass fluctuation kinetics (MFK), a set of coupled first-order differential equations that deterministically approximate the evolution of first and second moments of the CME by ignoring central moments of higher order.

Chapter 2

Mass Fluctuation Kinetics (MFK)

2.1 Introduction

The mass fluctuation kinetics (MFK) model approximates the evolution of first and second moments of the chemical master equation (CME) by ignoring central moments of higher order [8][9]. Because it incorporates the effects of second moments on first moments and vice versa, it is capable of capturing system dynamics better than MAK equations, which only track the first moments. MFK is a deterministic model and is thus significantly less computationally intensive than Monte Carlo methods.

This chapter is devoted to arriving at the MFK equations through a different approach from that presented in the original MFK papers. Specifically, the derivation presented here uses the temporal evolution of the molecule number moment generating function obtained from the CME to arrive at the equations. The original motivation for this approach was to provide, in the view of this author, a more transparent derivation, as well as to explore the possibility of systematically extending it to specific higher-order moments and studying the associated effects on system behavior. A paper currently in press (earlier than but independent from the work presented here) obtains a general analytic result for computing moment equations up to any order [1][1] using a similar

[1] Many thanks to Dr. Colin Gillespie for providing a prepublication version of this paper.

approach. The results here differ in emphasis and some details of development.

2.2 Preliminaries

For notational convenience, the time dependence of $x(t)$, $y(t)$ and $P(x,t)$ is henceforth made implicit, with the variables respectively referred to as x, y and $P(x)$. It should be recalled that the variables stand for the number of molecules at the current time t, the corresponding concentration and the probability of the system being in this state, respectively. The CME is restated here for convenience as

$$\frac{dP(x)}{dt} = \sum_{l=1}^{L} \left[P(x - s_l) a_l(x - s_l) - P(x) a_l(x) \right] \tag{2.1}$$

The mean concentration vector is defined as

$$\mu = E[y] = \frac{1}{\Omega} E[x] \in \mathbb{R}^n \tag{2.2}$$

and the concentration covariance matrix as

$$\mathbf{V} = \frac{1}{\Omega^2} E[(x - \mu)\,(x - \mu)^T] \in \mathbb{R}^{n \times n} \tag{2.3}$$

Equations (2.2) and (2.3) contain the state variables for MFK. For application purposes, it is convenient to rewrite the microscopic reaction rates, defined as

$$\left\{ \rho_l(y) \triangleq \frac{1}{\Omega} a_l(y\Omega) \right\}_{l=1}^{L} \tag{2.4}$$

in terms of the reaction rate constants $\{k_l\}_{l=1}^{L}$ that are readily available from reaction specifications, thereby substituting out the physical rate constants $\{c_l\}_{l=1}^{L}$, as shown in Table 2.1.

For the reaction categories being considered, each $\rho_l(y)$ may be expressed in quadratic

	Reaction category	$\rho_l(y)$
1.	zero-order reaction	k_l
2.	first-order reaction	$k_l y_i$
3.	homogeneous second-order reaction	$\frac{k_l}{\Omega} y_i \left(y_i \Omega - 1\right)$
4.	heterogeneous second-order reaction	$k_l y_i y_j \qquad i \neq j$

Table 2.1: *Propensity functions expressed in terms of reaction rate constants.*

form as

$$\rho_l(y) = k_l \left(b_l + \mathbf{c}_l^T y + y_l^T \mathbf{D}_l y\right) \tag{2.5}$$

with the parameters b_l, $\mathbf{c}_l$ and $\mathbf{D}_l$ collectively named the *microscopic rate parameters.*

2.3 Moment generating function evolution

The *moment generating function* for the molecule number vector $x \in \mathbb{R}^n$ is defined as

$$\mathcal{M}_x(v) = E\left[e^{v^T x}\right] = \sum_{\mathbf{x} \in \mathbb{Z}^n} \left[e^{v^T \mathbf{x}} P(\mathbf{x})\right] \tag{2.6}$$

where $v \in \mathbb{R}^n$ and the definition of E, as the *expectation* operator over x, has been made explicit. The temporal evolution equation for $\mathcal{M}_x(v)$ may be found by multiplying the CME by $e^{v^T x}$ and summing the result over all possible values of x, yielding

$$\frac{d\mathcal{M}_x(v)}{dt} = \sum_{l=1}^{L} \left[\sum_{\mathbf{x} \in \mathbb{Z}^n} \left[e^{v^T \mathbf{x}} P(\mathbf{x} - s_l) a_l(\mathbf{x} - s_l) - e^{v^T \mathbf{x}} P(\mathbf{x}) a_l(\mathbf{x})\right]\right] \tag{2.7}$$

This may be simplified further as

$$\frac{d\mathcal{M}_x(v)}{dt} = \sum_{l=1}^{L} \left[e^{v^T s_l} \sum_{\mathbf{x} \in \mathbb{Z}^n} \left[e^{v^T (\mathbf{x} - s_l)} P(\mathbf{x} - s_l) a_l(\mathbf{x} - s_l)\right] - \sum_{\mathbf{x} \in \mathbb{Z}^n} \left[e^{v^T \mathbf{x}} P(\mathbf{x}) a_l(\mathbf{x})\right]\right] \tag{2.8}$$

Recognizing that

$$\sum_{\mathbf{x}\in\mathbb{Z}^n}\left[e^{v^T\mathbf{x}}P(\mathbf{x})a_l(\mathbf{x})\right]=\sum_{\mathbf{x}\in\mathbb{Z}^n}\left[e^{v^T(\mathbf{x}-s_l)}P(\mathbf{x}-s_l)a_l(\mathbf{x}-s_l)\right] \tag{2.9}$$

(which may be verified by recognizing that the *support* of $x-s_l$ is just a shifted version of that of x, and is a subset of $\mathbb{Z}^n$) allows rewriting the moment generating function evolution equation as

$$\frac{d\mathcal{M}_x(v)}{dt}=\sum_{l=1}^{L}\left[\left(e^{v^Ts_l}-1\right)\sum_{\mathbf{x}\in\mathbb{Z}^n}\left[e^{v^T\mathbf{x}}P(\mathbf{x})a_l(\mathbf{x})\right]\right] \tag{2.10}$$

Deriving evolution equations for all moments from the equation above is a systematic task.

2.4 Mean evolution

The mean concentration vector μ may be generated from $\mathcal{M}_x(v)$ as

$$\mu=\frac{1}{\Omega}\left[\frac{d\mathcal{M}_x(v)}{dv}\right]_{v=0}=\frac{1}{\Omega}E\left[xe^{v^Tx}\right]_{v=0}=\frac{1}{\Omega}E[x] \tag{2.11}$$

Thus, the evolution equation for μ may be generated from Equation (2.10) as

$$\begin{aligned}\frac{d\mu}{dt}&=\frac{1}{\Omega}\left[\frac{d}{dv}\left[\frac{d\mathcal{M}_x(v)}{dt}\right]\right]_{v=0}\\&=\frac{1}{\Omega}\sum_{l=1}^{L}\left[\begin{array}{c}s_le^{v^Ts_l}\sum_{\mathbf{x}\in\mathbb{Z}^n}\left[e^{v^Tx}P(\mathbf{x})a_l(\mathbf{x})\right]\\+\left(e^{v^Ts_l}-1\right)\sum_{\mathbf{x}\in\mathbb{Z}^n}\left[\mathbf{x}e^{v^T\mathbf{x}}P(\mathbf{x})a_l(\mathbf{x})\right]\end{array}\right]_{v=0}\\&=\frac{1}{\Omega}\sum_{l=1}^{L}\left[s_l\sum_{\mathbf{x}\in\mathbb{Z}^n}\left[P(\mathbf{x})a_l(\mathbf{x})\right]\right]\\&=\sum_{l=1}^{L}s_lE\left[\rho_l(y)\right]\end{aligned} \tag{2.12}$$

This can be simplified further by observing that

$$E\left[\rho_l(y)\right] = E[k_l\left(b_l + \mathbf{c}_l^T y + y_l^T \mathbf{D}_l y\right)] = k_l\left(b_l + \mathbf{c}_l^T \mu + E\left[y_l^T \mathbf{D}_l y\right]\right) \tag{2.13}$$

and using the identity [10]

$$E\left[y_l^T \mathbf{D}_l y\right] = Tr(\mathbf{D}_l \mathbf{V}) + \mu_l^T \mathbf{D}_l \mu \tag{2.14}$$

to obtain

$$\frac{d\mu}{dt} = \sum_{l=1}^{L} \left\{ s_l k_l \left[b_l + \mathbf{c}_l^T \mu + \mu_l^T \mathbf{D}_l \mu + Tr(\mathbf{D}_l \mathbf{V}) \right] \right\} \tag{2.15}$$

Define the *average rate* for reaction R_l as its propensity function evaluated at μ, $\rho_l(\mu)$, its *stochastic rate* as

$$\xi_l(\mathbf{V}) \overset{\triangle}{=} k_l Tr(\mathbf{D}_l \mathbf{V}) = k_l vec\left\{\mathbf{D}_l\right\}^T vec\left\{\mathbf{V}\right\} \tag{2.16}$$

and its *effective rate* as the sum of its stochastic and average rates as

$$r_l \overset{\triangle}{=} \rho_l(\mu) + \xi_l(\mathbf{V}) = E\left[\rho_l(y)\right] \tag{2.17}$$

Defining the corresponding rate vectors as

$$\left\{ \rho = \begin{bmatrix} \rho_1(\mu) \\ \vdots \\ \rho_L(\mu) \end{bmatrix}; \quad \xi = \begin{bmatrix} \xi_1(\mathbf{V}) \\ \vdots \\ \xi_L(\mathbf{V}) \end{bmatrix}; \quad \mathbf{r} = \begin{bmatrix} r_1 \\ \vdots \\ r_L \end{bmatrix} \right\} \in \mathbb{R}^n$$

and recalling the definition of the stoichiometry matrix allows rewriting Equation (2.15) as

$$\frac{d\mu}{dt} = \mathbf{S}\mathbf{r} = \mathbf{S}(\rho + \xi) \tag{2.18}$$

This is the mean evolution equation. Note that defining the matrices

$$\mathcal{D} = \left[\begin{array}{ccc} vec(\mathbf{D}_1)| & ... & |vec(\mathbf{D}_L) \end{array}\right] \tag{2.19}$$

$$\mathbf{K} = diag\left\{\begin{array}{ccc} k_1 & \cdots & k_L \end{array}\right\} \tag{2.20}$$

$$\mathbf{C} = \left[\begin{array}{ccc} c_1| & \cdots & |c_L \end{array}\right] \tag{2.21}$$

allows writing the effective rate $\mathbf{r}$ in compact form as

$$\mathbf{r} = \mathbf{K}\left(b + \mathbf{C}^T\mu + \mathcal{D}^T(\mu \otimes \mu) + \mathcal{D}^T vec\{\mathbf{V}\}\right) \tag{2.22}$$

2.5 Covariance evolution

The covariance $\mathbf{V}$ may be generated from the *central moment generating function* $\mathcal{M}_{x-\mu}(v)$ as

$$\mathbf{V} = \frac{1}{\Omega^2}\left[\frac{d^2\mathcal{M}_{x-\mu\Omega}(v)}{dv^T dv}\right]_{v=0} = \frac{1}{\Omega^2}E\left[(x-\mu\Omega)(x-\mu\Omega)^T\right] \tag{2.23}$$

To derive the evolution equation for $\mathbf{V}$, we accordingly modify Equation (2.10) to obtain the central moment generating function evolution equation as

$$\frac{d\mathcal{M}_{x-\mu\Omega}(v)}{dt} = \sum_{l=1}^{L}\left[\left(e^{v^T s_l} - 1\right)\sum_{\mathbf{x}\in\mathbb{Z}^n}\left[e^{v^T(\mathbf{x}-\mu\Omega)}P(\mathbf{x})a_l(\mathbf{x})\right]\right] \tag{2.24}$$

This equation is readily obtained by replacing the term $e^{v^T\mathbf{x}}$ with $e^{v^T(\mathbf{x}-\mu\Omega)}$ in the steps leading up to the derivation of Equation (2.10). Thus, the evolution equation for $\mathbf{V}$ may be generated from Equation (2.24) as

$$
\begin{aligned}
\frac{d\mathbf{V}}{dt} &= \frac{1}{\Omega^2}\left[\frac{d^2}{dv^T dv}\left[\frac{d\mathcal{M}_{x-\mu\Omega}(v)}{dt}\right]\right]_{v=0} \\
&= \frac{1}{\Omega^2}\sum_{l=1}^{L}\left[\frac{d}{dv^T}\left[\begin{array}{l} s_l e^{v^T s_l}\sum\limits_{x=\mathbf{x}}\left[e^{v^T(\mathbf{x}-\mu\Omega)}P(\mathbf{x})a_l(\mathbf{x})\right] \\ +\left(e^{v^T s_l}-1\right)\sum\limits_{x=\mathbf{x}}\left[(\mathbf{x}-\mu\Omega)\,e^{v^T(\mathbf{x}-\mu\Omega)}P(\mathbf{x})a_l(\mathbf{x})\right] \end{array}\right]\right]_{v=0} \\
&= \frac{1}{\Omega^2}\sum_{l=1}^{L}\left[\begin{array}{l} s_l s_l^T e^{v^T s_l}\sum\limits_{\mathbf{x}\in\mathbb{Z}^n}\left[e^{v^T(\mathbf{x}-\mu\Omega)}P(\mathbf{x})a_l(\mathbf{x})\right] \\ +s_l e^{v^T s_l}\sum\limits_{\mathbf{x}\in\mathbb{Z}^n}\left[(\mathbf{x}-\mu\Omega)^T e^{v^T(\mathbf{x}-\mu\Omega)}P(\mathbf{x})a_l(\mathbf{x})\right] \\ +\left(\sum\limits_{\mathbf{x}\in\mathbb{Z}^n}\left[(\mathbf{x}-\mu\Omega)\,e^{v^T(\mathbf{x}-\mu\Omega)}P(\mathbf{x})a_l(\mathbf{x})\right]\right)s_l^T e^{v^T s_l} \\ +\left(e^{v^T s_l}-1\right)\sum\limits_{\mathbf{x}\in\mathbb{Z}^n}\left[(\mathbf{x}-\mu)\,(\mathbf{x}-\mu)^T e^{v^T(\mathbf{x}-\mu\Omega)}P(\mathbf{x})a_l(\mathbf{x})\right] \end{array}\right]_{v=0} \\
&= \frac{1}{\Omega^2}\sum_{l=1}^{L}\left[\begin{array}{l} s_l s_l^T\sum\limits_{\mathbf{x}\in\mathbb{Z}^n}\left[P(\mathbf{x})a_l(\mathbf{x})\right] \\ +s_l\sum\limits_{\mathbf{x}\in\mathbb{Z}^n}\left[(\mathbf{x}-\mu\Omega)^T P(\mathbf{x})a_l(\mathbf{x})\right] \\ +\left(\sum\limits_{\mathbf{x}\in\mathbb{Z}^n}\left[(\mathbf{x}-\mu\Omega)\,P(\mathbf{x})a_l(\mathbf{x})\right]\right)s_l^T \end{array}\right]
\end{aligned}
$$

$$
= \sum_{l=1}^{L}\left[\frac{1}{\Omega}s_l s_l^T r_l + s_l\left(E\left[y^T\rho_l(y)\right]-\mu^T r_l\right)+\left(E\left[y\rho_l(y)\right]-\mu r_l\right)s_l^T\right] \tag{2.25}
$$

To complete the derivation, an expression for the term $E\left[y\rho_l(y)\right] = E\left[y^T\rho_l(y)\right]^T$ needs to be found. This term can be simplified as

$$
\begin{aligned}
E\left[y\rho_l(y)\right] &= E[k_l\left(b_l y + y\mathbf{c}_l^T y + y y_l^T\mathbf{D}_l y\right)] \\
&= k_l\left(b_l\mu + E\left[y\mathbf{c}_l^T y\right] + E\left[y y_l^T\mathbf{D}_l y\right]\right)
\end{aligned} \tag{2.26}
$$

To proceed further, the following identity [10] is useful:

$$
E\left[y\left(\mathbf{c}_l^T y\right)\right] = E\left[y\left(y^T\mathbf{c}_l\right)\right] = \left(\mathbf{V}+\mu\mu^T\right)\mathbf{c}_l \tag{2.27}
$$

Also, when third central moments can be ignored [10],

$$\begin{aligned} E\left[yy_l^T\mathbf{D}_l y\right] &= Tr\left\{\mathbf{D}_l\mathbf{V}\right\}\mu + \mathbf{V}\mathbf{D}_l\mu + \mathbf{V}\mathbf{D}_l\mu + \mu\mu^T\mathbf{D}_l\mu \\ &= Tr\left\{\mathbf{D}_l\mathbf{V}\right\}\mu + 2\mathbf{V}\mathbf{D}_l\mu + \mu\mu^T\mathbf{D}_l\mu \end{aligned} \tag{2.28}$$

Then,

$$E\left[y\rho_l(y)\right] = k_l\left(b_l\mu + \left(\mathbf{V} + \mu\mu^T\right)\mathbf{c}_l + Tr\left\{\mathbf{D}_l\mathbf{V}\right\}\mu + 2\mathbf{V}\mathbf{D}_l\mu + \mu\mu^T\mathbf{D}_l\mu\right) \tag{2.29}$$

and

$$\begin{aligned} E\left[y\rho_l(y)\right] - \mu r_l &= k_l\left(b_l\mu + \left(\mathbf{V} + \mu\mu^T\right)\mathbf{c}_l + Tr\left\{\mathbf{D}_l\mathbf{V}\right\}\mu + 2\mathbf{V}\mathbf{D}_l\mu + \mu\mu^T\mathbf{D}_l\mu\right) \\ &- k_l\left(b_l + \mathbf{c}_l^T\mu + \mu_l^T\mathbf{D}_l\mu + Tr(\mathbf{D}_l\mathbf{V})\right)\mu \\ &= k_l\left(\mathbf{V}\mathbf{c}_l + 2\mathbf{V}\mathbf{D}_l\mu\right) \end{aligned} \tag{2.30}$$

Hence

$$\begin{aligned} \frac{d\mathbf{V}}{dt} &= \sum_{l=1}^{L}\left[\frac{1}{\Omega}s_l s_l^T r_l + s_l k_l\left(\mathbf{c}_l^T\mathbf{V} + 2\mu^T\mathbf{D}_l\mathbf{V}\right) + k_l\left(\mathbf{V}\mathbf{c}_l + 2\mathbf{V}\mathbf{D}_l\mu\right)s_l^T\right] \\ &= \frac{1}{\Omega}\mathbf{S\Lambda S}^T + \sum_{l=1}^{L}\left[s_l k_l\left(\mathbf{c}_l^T + 2\mu^T\mathbf{D}_l\right)\mathbf{V} + \mathbf{V}\left(\mathbf{c}_l + 2\mathbf{D}_l\mu\right)k_l s_l^T\right] \\ &= \frac{1}{\Omega}\mathbf{S\Lambda S}^T + \left(\mathbf{SKC}^T + 2\sum_{l=1}^{L}k_l s_l\mu^T\mathbf{D}_l\right)\mathbf{V} \\ &+\mathbf{V}\left(\mathbf{C}^T\mathbf{KS}^T + 2\sum_{l=1}^{L}k_l\mathbf{D}_l\mu s_l^T\right) \end{aligned} \tag{2.31}$$

Defining the *fluctuation dynamics matrix* $\mathbf{M}$ as

$$\begin{aligned}
\mathbf{M} &= \mathbf{SKC}^T + 2\sum_{l=1}^{L} k_l s_l \mu^T \mathbf{D}_l \\
&= \mathbf{SKC}^T + 2\mathbf{SK}\begin{bmatrix} \left(vec\left\{\mu^T\mathbf{D}_1\right\}\right)^T \\ \vdots \\ \left(vec\left\{\mu^T\mathbf{D}_L\right\}\right)^T \end{bmatrix} \\
&= \mathbf{SKC}^T + 2\mathbf{SK}\begin{bmatrix} \left(vec\left\{\mathbf{D}_1\right\}\right)^T (I_n \otimes \mu) \\ \vdots \\ \left(vec\left\{\mathbf{D}_L\right\}\right)^T (I_n \otimes \mu) \end{bmatrix} \\
&= \mathbf{SK}\left(\mathbf{C}^T + 2\mathcal{D}^T(I_n \otimes \mu)\right) \qquad (2.32)
\end{aligned}$$

allows rewriting Equation (2.31) as

$$\frac{d\mathbf{V}}{dt} = \mathbf{MV} + \mathbf{VM}^T + \frac{1}{\Omega}\mathbf{S\Lambda S}^T \qquad (2.33)$$

2.6 Summary and future work

Equations (2.12) and (2.33) define the MFK model. The MFK equations have thus been shown to follow from the molecule number moment generating function evolution equation that is obtained from the CME. The most pressing extension of this task, in the view of this author, is the systematic extension of the approach to specific higher-order moments and the study of associated effects on system behavior. In preparation for Chapters 4 and 5, note that $\{\Omega, \mathbf{V}\} = \{\infty, \mathbf{0}\}$ is an equilibrium MFK setting for $\mathbf{V}$, since Equation (2.33) is zero at that point. A quick look at Equations (2.12), (2.16), (2.17), (2.4) and (1.14) reveals that this MFK setting captures MAK dynamics.

Chapter 3

MFK Jacobian Matrix

3.1 Introduction

The goal of answering various questions about the local dynamics of the mass fluctuation kinetics (MFK) model necessitates establishing an analytic expression for its Jacobian matrix. This matrix contains the first-order derivatives of the MFK evolution equations with respect to each state variable. It is particularly useful in analyzing local stability properties of equilibria. If the real part of any Jacobian eigenvalue at an equilibrium is positive, the equilibrium is locally unstable. In particular, if this is true for all eigenvalues, the equilibrium is classified as a *repellor*, whereas when this applies to some but not all eigenvalues the equilibrium is a *saddle point.* Local stability of an equilibrium may be inferred from observing that eigenvalues have strictly negative real parts, classifying the equilibrium as an *attractor.* The Jacobian is also used in a variety of algorithms, among them the popular *Newton-Raphson* algorithm for computing equilibria.

This chapter establishes an analytic expression for the MFK Jacobian matrix, with the aid of several matrix results presented in Appendix A.

3.2 The vectorized model

To find the Jacobian, it is convenient to first vectorize the MFK model as

$$\begin{aligned}
vec\left\{\frac{d\mu}{dt}\right\} &= \frac{d\mu}{dt} = \mathbf{S}\mathbf{r} = \mathbf{S}(\rho+\xi) && (3.1)\\
vec\left\{\frac{d\mathbf{V}}{dt}\right\} &= vec\left\{\mathbf{M}\mathbf{V}+\mathbf{V}\mathbf{M}^T\right\} + vec\left\{\frac{1}{\Omega}\mathbf{S}\Lambda\mathbf{S}^T\right\} \\
&= (\mathbf{I}_n\otimes\mathbf{M}+\mathbf{M}\otimes\mathbf{I}_n)vec(\mathbf{V}) + \frac{1}{\Omega}(\mathbf{S}\otimes\mathbf{S})vec\left\{\Lambda\right\} \\
&= (\mathbf{I}_n\otimes\mathbf{M}+\mathbf{M}\otimes\mathbf{I}_n)F_n vech\{\mathbf{V}\} + \frac{1}{\Omega}(\mathbf{S}*\mathbf{S})\mathbf{r} \\
\Rightarrow \frac{d\left\{vech\left\{\mathbf{V}\right\}\right\}}{dt} &= E_n(\mathbf{I}_n\otimes\mathbf{M}+\mathbf{M}\otimes\mathbf{I}_n)F_n vech\{\mathbf{V}\} + \frac{1}{\Omega}E_n(\mathbf{S}*\mathbf{S})\mathbf{r} && (3.2)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{r} &= \mathbf{K}\left(b + C^T\mu + \mathcal{D}^T(\mu\otimes\mu) + \mathcal{D}^T vec\{\mathbf{V}\}\right) \\
&= \mathbf{K}\left(b + C^T\mu + \mathcal{D}^T(\mu\otimes\mu) + \mathcal{D}^T F_n vech\{\mathbf{V}\}\right) && (3.3)\\
\mathbf{M} &= \mathbf{S}\mathbf{K}\left(C^T + 2\mathcal{D}^T(I_n\otimes\mu)\right) \in \mathbb{R}^{n\times n} && (3.4)
\end{aligned}$$

The state vector for the vectorized MFK model as described above is

$$z = \begin{bmatrix} \mu \\ \mathbf{V}_{vech} \end{bmatrix} \in \mathbb{R}^{\frac{n(n+3)}{2}} \qquad (3.5)$$

where $\mathbf{V}_{vech}$ denotes $vech\left\{\mathbf{V}\right\}$. Let the vectorized MFK model be denoted as

$$\frac{dz}{dt} = g(z) = \begin{bmatrix} \frac{d\mu}{dt} \\ \frac{d\mathbf{V}_{vech}}{dt} \end{bmatrix} \in \mathbb{R}^{\frac{n(n+3)}{2}} \qquad (3.6)$$

3.3 The Jacobian matrix

The Jacobian for the vectorized model is defined as

$$\mathbf{J} = \frac{dg(z)}{dz} = \left[\begin{array}{cc} \frac{dg(z)}{d\mu} & \frac{dg(z)}{d\mathbf{V}_{vech}} \end{array} \right] \in \mathbb{R}^{\frac{n(n+3)}{2} \times \frac{n(n+3)}{2}} \tag{3.7}$$

This may be written more explicitly as

$$\mathbf{J} = \left[\begin{array}{cc} \frac{d}{d\mu}\left\{\frac{d\mu}{dt}\right\} & \frac{d}{d\mathbf{V}_{vech}}\left\{\frac{d\mu}{dt}\right\} \\ \frac{d}{d\mu}\left\{\frac{d\mathbf{V}_{vech}}{dt}\right\} & \frac{d}{d\mathbf{V}_{vech}}\left\{\frac{d\mathbf{V}_{vech}}{dt}\right\} \end{array} \right] \tag{3.8}$$

Hence, finding the Jacobian just amounts to finding the four terms in the block matrix above. Three of the four terms are found as

$$\begin{aligned}
\frac{d}{d\mu}\left\{\frac{d\mu}{dt}\right\} &= \frac{d}{d\mu}\{\mathbf{Sr}\} = \mathbf{SK}\left(C^T + 2\mathcal{D}^T(\mathbf{I}_n \otimes \mu)\right) = \mathbf{M} && (3.9)\\
\frac{d}{d\mathbf{V}_{vech}}\left\{\frac{d\mu}{dt}\right\} &= \frac{d}{d\mathbf{V}_{vech}}\{\mathbf{Sr}\} = \mathbf{SK}\mathcal{D}^T F_n && (3.10)\\
\frac{d}{d\mathbf{V}_{vech}}\left\{\frac{d\mathbf{V}_{vech}}{dt}\right\} &= \frac{d}{d\mathbf{V}_{vech}}\{E_n(\mathbf{I}_n \otimes \mathbf{M} + \mathbf{M} \otimes \mathbf{I}_n)F_n\mathbf{V}_{vech}\} && (3.11)\\
&\quad + \frac{d}{d\mathbf{V}_{vech}}\left\{\frac{1}{\Omega}E_n(\mathbf{S} * \mathbf{S})\mathbf{r}\right\} && (3.12)\\
&= E_n(I_n \otimes \mathbf{M} + \mathbf{M} \otimes I_n)F_n + \frac{1}{\Omega}E_n(\mathbf{S} * \mathbf{S})\mathbf{K}\mathcal{D}^T F_n && (3.13)
\end{aligned}$$

Finding the last term is more complicated. The expression needs to be first rearranged and simplified to ensure that dimensions remain correct:

$$\begin{aligned}
\frac{d}{d\mu}\left\{\frac{d\mathbf{V}_{vech}}{dt}\right\} &= \frac{d}{d\mu}\left\{E_n(I_n \otimes \mathbf{M} + \mathbf{M} \otimes I_n)F_n\mathbf{V}_{vech} + \frac{1}{\Omega}E_n(\mathbf{S} * \mathbf{S})\mathbf{r}\right\}\\
&= \frac{d}{d\mu}\left\{E_n(vec\{\mathbf{MV}\} + vec\left\{\mathbf{VM}^T\right\}) + \frac{1}{\Omega}E_n(\mathbf{S} * \mathbf{S})\mathbf{r}\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{d}{d\mu}\left\{E_n(\mathbf{V}\otimes I_n + (I_n\otimes\mathbf{V})\,\mathbf{P}_{nn})vec\{\mathbf{M}\}\right\} \\
&\quad + \frac{1}{\Omega}E_n(\mathbf{S}*\mathbf{S})\mathbf{K}\left(C^T + 2\mathcal{D}^T(I_n\otimes\mu)\right) \qquad (3.14)\\
&= E_n(\mathbf{V}\otimes I_n + (I_n\otimes\mathbf{V})\,\mathbf{P}_{nn})\frac{d}{d\mu}\left\{vec\left\{2\mathbf{SK}\mathcal{D}^T(I_n\otimes\mu)\right\}\right\} \qquad (3.15)\\
&\quad + \frac{1}{\Omega}E_n(\mathbf{S}*\mathbf{S})\mathbf{K}\left(C^T + 2\mathcal{D}^T(I_n\otimes\mu)\right)\\
&= E_n(\mathbf{V}\otimes I_n + (I_n\otimes\mathbf{V})\,\mathbf{P}_{nn})\left(I_n\otimes 2\mathbf{SK}\mathcal{D}^T\right)\frac{d}{d\mu}\left\{vec\{I_n\otimes\mu\}\right\}\\
&\quad + \frac{1}{\Omega}E_n(\mathbf{S}*\mathbf{S})\mathbf{K}\left(C^T + 2\mathcal{D}^T(I_n\otimes\mu)\right) \qquad (3.16)
\end{aligned}
$$

Finding the last unknown expression involves thinking about the definitions of the *vec* and $\otimes$ operators and performing further rearrangements as

$$
\begin{aligned}
\frac{d}{d\mu}\left\{vec\{I_n\otimes\mu\}\right\} &= \frac{d}{d\mu}\left\{vec\left\{\begin{bmatrix}\mu & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \mu\end{bmatrix}\right\}\right\}\\
&= \frac{d}{d\mu}\left\{vec\left\{\begin{bmatrix}1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 1\end{bmatrix}\right\}\otimes\mu\right\}\\
&= vec\left\{\begin{bmatrix}1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 1\end{bmatrix}\right\}\otimes\left(\frac{d}{d\mu}\{\mu\}\right)\\
&= vec\{I_n\}\otimes I_n \qquad (3.17)\\
\Rightarrow \frac{d}{d\mu}\left\{\frac{d\mathbf{V}_{vech}}{dt}\right\} &= E_n\left\{\mathbf{V}\otimes I_n + (I_n\otimes\mathbf{V})\,\mathbf{P}_{nn}\right\}\left(I_n\otimes 2\mathbf{SK}\mathcal{D}^T\right)(vec\{I_n\}\otimes I_n)\\
&\quad + \frac{1}{\Omega}E_n(\mathbf{S}*\mathbf{S})\mathbf{K}\left(C^T + 2\mathcal{D}^T(I_n\otimes\mu)\right) \qquad (3.18)
\end{aligned}
$$

Equations (3.9), (3.10), (3.13) and (3.18) define the MFK Jacobian matrix.

Chapter 4

MFK Software Prototype

4.1 Introduction

Our interest in exploring various properties of the mass fluctuation kinetics (MFK) model makes the development of an MFK software toolbox appropriate. A software prototype has been developed in MATLAB, a popular numerical computing environment, to automate the analysis of systems of chemical reactions within the framework. It is available to download at `http://www.mit.edu/~azunre/MFK` (note that MATLAB and its `SimBiology` and `Symbolic Math` toolboxes are required to run it). The current version performs the basic functions of describing the model, finding equilibria and studying the variation of local dynamics with varying *system size*. This chapter reviews the technical specifications of the software prototype and the theory behind its current capabilities.

4.2 Technical specifications

The prototype was developed with the aid of MATLAB's `SimBiology` and `Symbolic Math` toolboxes. `SimBiology` allows model description via a Graphical User Interface (GUI) and provides routines for all algorithms that have been discussed in this text so far, such as Gillespie's stochastic simulation algorithm (SSA). It is also capable of interpreting models described in the Systems Biology Markup Language (SBML), a widely used standard for encoding such models. Thus, it may potentially be applied to numerous SBML models readily available online. The Symbolic Math toolbox allows handling expressions in symbolic form, which enables their human-readable display in many cases. This may be useful in establishing analytic conclusions about specific aspects of system behavior without having to explicitly perform all of the supporting algebra.

A system to be studied must be first described, along with its initial conditions and key parameter settings, via the SimBiology GUI (or an SBML model opened with the GUI) and saved under a desired name in SimBiology's .sbproj format (SimBiology documentation includes a thorough description of this procedure). The path to the model must then be provided as input to function PrepareModel, which prepares its MFK model description and stores it in the MATLAB workspace. Subsequently, any function may be called on to manipulate this description. The beginning of each function file provides detailed instructions on use and a brief description of its capabilities. The theory behind each currently available capability is provided in the next section. Chapter 5 applies these capabilities to specific examples.

4.3 Capabilities and supporting theory

4.3.1 Preparing and displaying the MFK description

The core functionality of the prototype involves preparing and storing the MFK description of a system in the MATLAB workspace, with the aid of function PrepareModel. This task theoretically entails finding non-redundant symbolic expressions for the MFK evolution equations and the associated Jacobian matrix. The procedure for eliminating redundancy of species used in describing a system, in slightly modified form from that described in [11] (see also [12]), is addressed next.

Moiety conservation analysis

The total amounts of some subgroups of species in systems of chemical reactions may be conserved, in which case not all species present are needed to describe their temporal evolution. These subgroups are termed *conserved moieties* and the time-invariant amount of each one depends on the initial conditions of its constituents. They are present if the stoichiometry matrix $\mathbf{S}$ is row-rank deficient. The rows of every $\mathbf{S}$ may be rearranged as

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_I \\ \mathbf{S}_D \end{bmatrix} \in \mathbb{R}^{n \times L} \tag{4.1}$$

which corresponds to rearranging the vector of species X into independent species X_I and dependent species X_D as

$$X = \begin{bmatrix} X_I \\ X_D \end{bmatrix} \tag{4.2}$$

Here, $\mathbf{S}_I$ is a full rank matrix of rows of $\mathbf{S}$ corresponding to the independent species in X_I, while $\mathbf{S}_D$ corresponds to the dependent species in X_D. Let the rank of $\mathbf{S}_I$, corresponding to the number of independent species in X_I, be denoted by n_I. Every row of $\mathbf{S}_D$ may be expressed as a linear combination of some rows of $\mathbf{S}_I$, as captured by the *link-zero matrix* $\mathbf{L}_0 \in \mathbb{R}^{(n-n_I)\times n_I}$ in

$$\mathbf{S}_D = \mathbf{L}_0 \mathbf{S}_I \tag{4.3}$$

Then, $\mathbf{S}$ may be written as

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_I \\ \mathbf{S}_D \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_I} \\ \mathbf{L}_0 \end{bmatrix} \mathbf{S}_I \tag{4.4}$$

Combining this with the definition of the stoichiometry matrix allows establishing the following relationship between the initial number of molecules, $x(0)$, and the number of molecules at time t, $x(t)$:

$$x(t) - x(0) = \begin{bmatrix} \mathbf{I}_{n_I} \\ \mathbf{L}_0 \end{bmatrix} \mathbf{S}_I m \tag{4.5}$$

Here, $m \in \mathbb{R}^L$ is a column vector representing the number of times each reaction fires in the time interval $[0, t]$. Equations (4.2) and (4.5) are combined as

$$\begin{aligned} \begin{bmatrix} x_I(t) - x_I(0) \\ x_D(t) - x_D(0) \end{bmatrix} &= \begin{bmatrix} \mathbf{I}_{n_I} \\ \mathbf{L}_0 \end{bmatrix} \mathbf{S}_I m \\ \Rightarrow x_D(t) - x_D(0) &= \mathbf{L}_0 \left[x_I(t) - x_I(0) \right] \\ \Rightarrow -\mathbf{L}_0 x_I(t) + x_D(t) &= -\mathbf{L}_0 x_I(0) + x_D(0) \end{aligned} \tag{4.6}$$

Defining the conservation matrix as

$$\mathbf{\Gamma} = \begin{bmatrix} -\mathbf{L}_0 & \mathbf{I}_{n_I} \end{bmatrix} \tag{4.7}$$

allows writing Equation (4.6) as

$$\mathbf{\Gamma} x(t) = \mathbf{\Gamma} x(0) \tag{4.8}$$

The subgroups of species whose molecule numbers/concentrations are conserved, the conserved moieties, are then given by $\mathbf{\Gamma} X$. Their time-invariant concentrations of interest are obtained from initial concentrations as

$$\pi = \mathbf{\Gamma} y(0) \tag{4.9}$$

The concentrations of dependent species may be expressed in terms of independent concentrations as

$$\begin{aligned} \pi = \mathbf{\Gamma} y(t) = \begin{bmatrix} -\mathbf{L}_0 & \mathbf{I}_{n_I} \end{bmatrix} \begin{bmatrix} y_I(t) \\ y_D(t) \end{bmatrix} \\ \Rightarrow y_D(t) = \pi + \mathbf{L}_0 y_I(t) \end{aligned} \tag{4.10}$$

Dependent concentrations may now be substituted out of all MFK evolution equations and system evolution described by independent concentrations only.

The `SimBiology` function `sbioconsmoiety` implements a number of algorithms for obtaining $\mathbf{\Gamma}$ and hence the time-invariant concentrations in π. An algorithm based on the QR-decomposition of $\mathbf{S}$, which is more efficient at handling large systems [11], is used by the software prototype. If necessity to perform moiety conservation analysis is detected, the user is asked to specify indices of independent species of interest as a row vector via the MATLAB command window.

Displaying the MFK description

The prototype enables display of symbolic expressions for the MFK evolution equations and the Jacobian matrix in human-readable form. This is not available for all systems, as in some cases (perhaps most practical cases), the expressions are too long to be displayed effectively (nothing is displayed in this case, so it doesn't hurt to try). To alleviate this drawback at the expense of desktop clutter, an option is included for each MFK evolution equation and each row of the Jacobian to be shown in separate windows. All symbolic expressions are based on the equations derived in Chapters 2 and 3, with appropriate parameters extracted from the `SimBiology` model description. In particular, the microscopic rate parameters are extracted from the MAK rate law for each reaction. The MFK state vector is also shown if the display option is enabled.

4.3.2 Plotting trajectories

The function `TrajectoryPlot` generates plots of MFK trajectories using initial conditions as specified in the `SimBiology` model description. The trajectories are obtained by integrating the MFK equations using MATLAB's `ode15s` ODE solver, capable of handling moderately stiff systems. Options are provided to simulate exact trajectories via the stochastic simulation algorithm [SSA] or the tau-leaping algorithm.

4.3.3 Finding equilibria via Newton-Raphson

Let the MFK state vector, which contains the means and covariances of the n_I independent species only, be denoted as $z \in \mathbb{R}^{\frac{n_I(n_I+3)}{2}}$. Let the vectorized MFK model be denoted as $\frac{dz}{dt} = g(z)$. An equilibrium $\bar{z}$ is characterized by the condition

$$g(\bar{z}) = \left[\frac{dz}{dt}\right]_{z=\bar{z}} = 0 \tag{4.11}$$

Hence, equilibria are simply the zeros of $g(z)$. An equilibrium may be found iteratively using the Newton-Raphson algorithm. At the i^{th} iteration, the definition of the Jacobian matrix $\mathbf{J}$ establishes the expression

$$[\mathbf{J}]_{z=z_i}(z_{i+1} - z_i) \approx g(z_{i+1}) - g(z_i) \tag{4.12}$$

Here, z_{i+1} is the next successively better approximation for the equilibrium and

$$[\mathbf{J}]_{z=z_i} = \left[\frac{dg(z)}{dz}\right]_{z=z_i} \tag{4.13}$$

is the Jacobian evaluated at z_i. The following iteration can then be established:

$$z_{i+1} = \left([\mathbf{J}]_{z=z_i}\right)^{-1}[g(z_{i+1}) - g(z_i)] + z_i \tag{4.14}$$

Starting at some initial guess z_0, iterations are performed until the fractional change in solution over one iteration, $\frac{\|z_{i+1}-z_i\|}{\|z_i\|}$, is below some user-specified convergence tolerance. Figure 4-1 visualizes two iterations of the algorithm in a toy one-dimensional example. The algorithm starts at z_0 and moves on to points z_1 and z_2, approaching the zero at $\bar{z}_1$. As is evident from the figure, the algorithm may find different equilibria depending on the initial guess and often can't find all of them (assuming equilibria exist). However, it provides sufficient results for the purposes of this text. The use of more sophisticated root-finding algorithms would be a worthwhile next development step for the software prototype. Function `FindEquilibrium` implements the Newton-Raphson algorithm on the model of interest, to a user-specified convergence tolerance. The initial guess is obtained from the initial concentration vector specified in the `SimBiology` model.

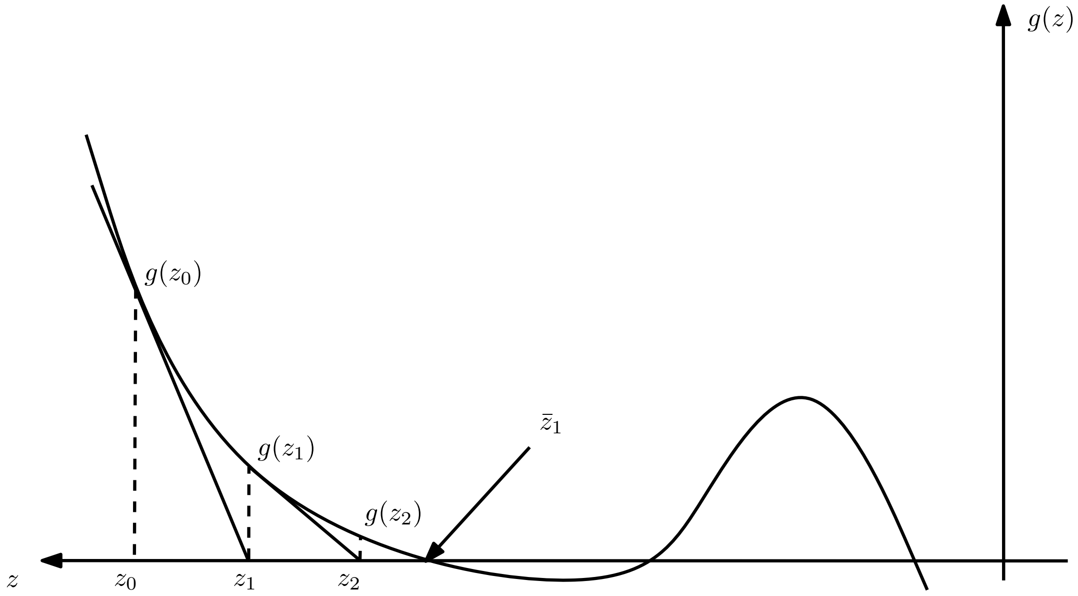


Figure 4-1: *Two iterations of the Newton Raphon algorithm on a simple example*

4.3.4 Local bifurcation analysis

Local bifurcation analysis studies the evolution of the local stability properties of an equilibrium with changes in a model parameter of interest. The parameter that readily comes to mind for MFK is Ω, since the MFK setting $\{\Omega, \mathbf{V}\} = \{\infty, \mathbf{0}\}$ captures MAK dynamics. The understanding of MFK as a modeling tool benefits from the exploration of its local bifurcation link to MAK, the point of reference for most of stochastic chemical kinetic theory. This exploration begins with finding the MAK equilibrium via Newton-Raphson at the MFK setting $\{\Omega, \mathbf{V}\} = \{\infty, \mathbf{0}\}$. Let this equilibrium be denoted as $\bar{z}_{MAK}$. Defining *inverse system size* as

$$\mho = \frac{1}{\Omega} \tag{4.15}$$

allows expressing this MFK setting as $\{\mho, \mathbf{V}\} = \{0, \mathbf{0}\}$. A differential equation for the evolution of the equilibrium with $\mho$ is established next. This is done by differentiating the vectorized MFK model $g(z, \mho)$, now considered as a function of both z and $\mho$, with respect to $\mho$ to obtain

$$\frac{d\left\{g(z,\mho)\right\}}{d\mho} = \begin{bmatrix} 0 \\ E_n(\mathbf{S} * \mathbf{S})\mathbf{r} \end{bmatrix} \tag{4.16}$$

A first-order Taylor expansion of the model around an arbitrary equilibrium $\bar{z}$, realized at $\mho = \bar{\mho}$, is performed as

$$\begin{aligned} g(\bar{z}, \bar{\mho}) &= 0 \\ \Rightarrow \left[\frac{d\left\{g(z,\mho)\right\}}{dz}\right]_{\left\{\bar{z},\bar{\mho}\right\}} \cdot \delta\bar{z} + \left[\frac{d\left\{g(z,\mho)\right\}}{d\mho}\right]_{\left\{\bar{z},\bar{\mho}\right\}} \cdot \delta\bar{\mho} &\approx 0 \\ \Rightarrow -\left([\mathbf{J}]_{z=\bar{z},\mho=\bar{\mho}}\right)^{-1} \begin{bmatrix} 0 \\ E_n(\mathbf{S} * \mathbf{S})\mathbf{r} \end{bmatrix} &\approx \frac{\delta\bar{z}}{\delta\bar{\mho}} \end{aligned} \tag{4.17}$$

In the limit of small perturbations $\delta\bar{z}$ and $\delta\bar{\mho}$, this expression becomes a differential equation for the evolution of the equilibrium with inverse system size, as can be seen from

$$\frac{d\bar{z}}{d\bar{\mho}} = \lim_{\left\{\delta\bar{z},\delta\bar{\mho}\right\}\longrightarrow 0} \left\{\frac{\delta\bar{z}}{\delta\bar{\mho}}\right\} = -\left([\mathbf{J}]_{z=\bar{z},\mho=\bar{\mho}}\right)^{-1} \begin{bmatrix} 0 \\ E_n(\mathbf{S} * \mathbf{S})\mathbf{r} \end{bmatrix} \tag{4.18}$$

Beginning at $\left\{\bar{z}, \bar{\mho}\right\} = \left\{\bar{z}_{MAK}, \mathbf{0}\right\}$, this equation is integrated with MATLAB's `ode15s` ODE solver, until some user-specified final value of $\mho$ (note that analysis may terminate early due to the Jacobian becoming singular). This yields a sequence of equilibria $\bar{z}$ corresponding to a sequence of $\bar{\mho}$ values. At each $\left\{\bar{z}, \bar{\mho}\right\}$ pair along this equilibrium evolution path, the Jacobian may be evaluated and its eigenvalues found. The evolution of the real and imaginary parts of the Jacobian eigenvalues provide an insight on the evolution of the local stability properties of the equilibrium. If the final $\mho$ value is large enough (Ω small enough), the MFK equations are expected to become unstable due to the $\frac{1}{\Omega}\mathbf{S}\mathbf{\Lambda}\mathbf{S}^T$ term in Equation (2.33), corresponding to a bifurcation (an attractor turning into a saddle or a repellor). Assuming that true system behavior remains stable, which is typically expected to be the case, a range outside which MFK is no longer a useful approximation for system behavior is thereby established.

Function `BifurcationPlot` generates eigenvalue evolution plots for a user-specified range of $\mho$ (beginning in the MAK regime $\left\{\bar{z}, \bar{\mho}\right\} = \{\bar{z}_{MAK}, \mathbf{0}\}$). The bifurcation may be visualized in the time domain by plotting MFK trajectories with `TrajectoryPlot` for $\mho$ at regularly sampled values in the interval of interest. A *bifurcation movie* may be generated by combining a specified number of trajectories as frames over a range of $\mho$, producing an animation that provides more intuition on the bifurcation. The script `BifurcationMovie`, included with the prototype download, may be used to generate bifurcation movies.

4.3.5 Local mean-covariance coupling analysis

Since the temporal evolution of means in μ at the MFK setting $\{\mho, \mathbf{V}\} = \{0, \mathbf{0}\}$ captures MAK dynamics, μ and covariances in $\mathbf{V}$ are locally decoupled at the MAK equilibrium $\bar{z}_{MAK}$. Evolution of the local coupling between μ and $\mathbf{V}$ at this equilibrium with $\mho$, as captured by *participation factor* evolution of the linearized MFK model along the equilibrium evolution path described in the previous subsection, provides further insight into the connection between MAK and MFK.

The MFK model linearized around a stable equilibrium $\bar{z}$ is given by the linear time invariant system

$$\frac{d\left\{\delta z\right\}}{dt} = \left[\mathbf{J}\right]_{z=\bar{z}} \cdot \delta z \tag{4.19}$$

Here, $\delta z \in \mathbb{R}^{n_v}$ with $n_v = \frac{n_I(n_I+3)}{2}$ stands for small perturbation in z from $\bar{z}$. If $\left[\mathbf{J}\right]_{z=\bar{z}}$ is diagonalizable (for which a sufficient though not necessary condition is that its eigenvalues $\{\lambda_q\}_{q=1}^{n_v}$ are distinct), then given some initial condition δz_0, a solution for the temporal state evolution of this system may be written as

$$\delta z = V e^{\Psi t} W \delta z_0 = \sum_{q=1}^{n_v} v_q w_q^T e^{\lambda_q t} \delta z_0 \tag{4.20}$$

Here, V and W are matrices with q^{th} column v_q and q^{th} row w_q^T equal to the q^{th} right eigenvector and q^{th} left eigenvector of $[\mathbf{J}]_{z=\bar{z}}$ respectively, and Ψ is a diagonal matrix of eigenvalues $\{\lambda_q\}_{q=1}^{n_v}$. This suggests defining the participation factor of the k^{th} perturbation $\{\delta z\}_k$ in the q^{th} mode λ_q as

$$p_{kq} = \{v_q\}_k \left\{w_q^T\right\}_k \tag{4.21}$$

See [13] and [14] for developments of this idea. Choosing $W = V^{-1}$ ensures that $\sum_{k=1}^{n_v} p_{kq} = 1$ and enables a matrix of participation factors with rows corresponding to δz and columns corresponding to eigenvalues $\{\lambda_q\}_{q=1}^{n_v}$ to be computed as

$$P = \left(V^{-1}\right)^T \odot V \tag{4.22}$$

Here $\odot$ denotes the Hadamard elementwise product. The relative magnitudes of participation factors highlight the relative participation of each state variable in each mode. Disjoint sets of state variables that participate in disjoint sets of modes are likely to be decoupled. When the participation of a state variable in a set of modes is low compared to other state variables, an opportunity for model reduction may be present.

Function `ParticipationFactorPlot` generates an evolution plot of the participation factors of a given state variable in each of the n_v modes along the equilibrium evolution path, beginning at $\bar{z}_{MAK}$ and ending at some user-specified value of $\mho$.

Chapter 5

MFK Enzyme Kinetics

5.1 Introduction

Enzyme kinetics is the study of chemical reactions that are mediated by the presence of chemical species known as *enzymes*. The underlying mechanism of action involves manipulation of molecules of other chemical species known as *substrates*. This chapter analyzes MFK enzyme kinetics in the presence of a single substrate. Analysis is facilitated by the software toolbox prototype described in Chapter 4. All files required to replicate the presented results are included with the prototype download, available at `http://www.mit.edu/~azunre/MFK`. Bifurcation movies for the three examples studied, which provide time-domain visualization of the local bifurcation analysis procedure central to this chapter, are also available on the companion website. The purpose of this chapter is to provide an example application of the model and software to specific systems, while attempting to further the understanding of local dynamics properties of MFK equilibria and their connection to MAK. This exploration is backed by analytic justification as much as possible, to enable it serve as a tutorial for anyone interested in studying the model and/or applying the software prototype. Note that parameter settings and initial conditions for computational analysis were selected by trial and error to illustrate varying points.

5.2 Single substrate enzyme reaction system

As the name suggests, the *single substrate enzyme reaction system* involves a single substrate S interacting with the enzyme E to form product P, via two stages. The first stage involves formation of a complex C through a reversible reaction written as

$$E + S \underset{k_2}{\overset{k_1}{\rightleftharpoons}} C \tag{5.1}$$

The forward reaction is denoted as R_1 and the backward reaction as R_2. This reversible reaction is referred to as *complex formation and dissociation* (CFD). During the second stage, C decomposes to form P and E. When the second stage is not reversible, the two stages are written as

$$E + S \underset{k_2}{\overset{k_1}{\rightleftharpoons}} C \xrightarrow{k_3} P + E \tag{5.2}$$

The second stage reaction is denoted as R_3. This is the *irreversible enzyme reaction system* (IERS). When the second stage is reversible, the two stages are written as

$$E + S \underset{k_2}{\overset{k_1}{\rightleftharpoons}} C \underset{k_4}{\overset{k_3}{\rightleftharpoons}} P + E \tag{5.3}$$

The forward reaction for the second stage is denoted as R_3 and the backward reaction as R_4. This is the *reversible enzyme reaction system* (RERS).

5.3 Analysis and discussion

5.3.1 Complex formation and dissociation (CFD)

Analytic expressions describing *complex formation and dissociation* MFK are derived explicitly and compared with corresponding toolbox output. The purpose of the treatment here is to confirm prototype output credibility while providing a thorough example application of MFK. Analysis of local dynamics is also presented.

Preliminaries

Stoichiometry vectors for R_1 and R_2, s_1 and s_2 respectively, are grouped into the stoichiometry matrix $\mathbf{S}$ as

$$\mathbf{S} = \left[\begin{array}{cc} s_1 & s_2 \end{array} \right] = \begin{array}{c|cc|} & R_1 & R_2 \\ E & -1 & 1 \\ S & -1 & 1 \\ C & 1 & -1 \end{array} \tag{5.4}$$

$\mathbf{S}$ may be verified by recalling that each of its entries indicates how many molecules of a species are produced (if entry positive) or consumed (if entry negative) by each reaction. The correspondence of rows to species E, S and C, and the correspondence of columns to reactions R_1 and R_2, has been made explicit. Thus, $\mathbf{S}$ corresponds to the species concentration vector

$$y = \begin{bmatrix} y_E \\ y_S \\ y_C \end{bmatrix} \tag{5.5}$$

Analytic expressions

Time-invariant concentrations of conserved moieties in this simple system may be obtained by inspection. One molecule of E is consumed by R_1 to produce one molecule of C, which is reversed exactly by R_2. Thus, the quantity

$$\pi_1 = y_E + y_C \tag{5.6}$$

must be conserved. Similar reasoning concludes that the quantity

$$\pi_2 = y_S + y_C \tag{5.7}$$

is conserved as well. Moiety conservation analysis, as described in Chapter 4, obtains conserved moieties in a systematic fashion that extends easily to systems of greater complexity. This procedure is applied next.

Since s_2 is a scalar multiple of s_1, $\mathbf{S}$ has rank one, containing one independent row (corresponding to an arbitrarily selected independent species) and two dependent rows (hence, two conserved moieties). If C is chosen as the one independent species of interest, the concentration vector may be rearranged as

$$y = \begin{bmatrix} y_I \\ \hline y_D \end{bmatrix} = \begin{bmatrix} y_C \\ \hline y_E \\ y_S \end{bmatrix} \tag{5.8}$$

and the concentrations of conserved moieties obtained as

$$\begin{aligned}
\mathbf{S}_I &= \begin{bmatrix} 1 & -1 \end{bmatrix} \\
\mathbf{S}_D &= \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} = \overbrace{\begin{bmatrix} -1 \\ -1 \end{bmatrix}}^{\mathbf{L}_0} \mathbf{S}_I \\
\Rightarrow \ \pi = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} &= \overbrace{\begin{bmatrix} -\mathbf{L}_0 & \mathbf{I}_2 \end{bmatrix}}^{\boldsymbol{\Gamma}} \overbrace{\begin{bmatrix} y_C \\ y_E \\ y_S \end{bmatrix}}^{y} = \begin{bmatrix} y_C + y_E \\ y_C + y_S \end{bmatrix}
\end{aligned} \tag{5.9}$$

This matches expressions obtained by inspection.

The MFK state vector after conservation analysis (containing means and covariances of independent concentrations in y_I only) is

$$z = \begin{bmatrix} E\left[y_I\right] \\ cov\left\{y_I, y_I\right\} \end{bmatrix} = \begin{bmatrix} \mu_C \\ \sigma_{CC} \end{bmatrix} \tag{5.10}$$

To clarify notation, it should be noted that

$$\sigma_{CC} = cov\left(y_C, y_C\right) = var\left(y_C\right) = \sigma_C^2 \tag{5.11}$$

To determine the MFK evolution equations, microscopic reaction rates are first written for R_1 (a heterogeneous second-order reaction) and R_2 (a first-order reaction) as

$$\begin{aligned}
\rho_1(y_I) &= k_1 y_E y_S = k_1\left(\pi_1 - y_C\right)\left(\pi_2 - y_C\right) \\
&= k_1\left[\pi_1\pi_2 - \left(\pi_1 + \pi_2\right)y_C + y_C^2\right] && (5.12) \\
\rho_2(y_I) &= k_2 y_C && (5.13)
\end{aligned}$$

The microscopic rate parameters may now be obtained by direct comparison with Equation (2.5) as

$$\begin{aligned}
& b_1 = \pi_1\pi_2 \\
& \mathbf{c}_1 = -\left(\pi_1 + \pi_2\right) \\
& \mathbf{D}_1 = 1 \\
& b_2 = 0 \\
& \mathbf{c}_2 = 1 \\
& \mathbf{D}_2 = 0
\end{aligned} \tag{5.14}$$

The MFK evolution equations may now be obtained via direct substitution of these parameters into Equations (2.12) and (2.33). The mean evolution equation is obtained as

$$\begin{aligned}
\frac{d\mu_C}{dt} &= \mathbf{S}_I\mathbf{r} = \mathbf{S}_I\mathbf{K}\left(b + C^T\mu + \mathcal{D}^T(\mu\otimes\mu) + \mathcal{D}^T F_1 vech\{\mathbf{V}\}\right) \\
&= \overbrace{\begin{bmatrix} 1 & -1 \end{bmatrix}}^{\mathbf{S}_I}\overbrace{\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}\begin{bmatrix} \pi_1\pi_2 - \left(\pi_1 + \pi_2\right)\mu_C + \mu_C^2 + \sigma_{CC} \\ \mu_C \end{bmatrix}}^{\mathbf{r}} \\
&= k_1\left[\pi_1\pi_2 - \left(\pi_1 + \pi_2\right)\mu_C + \mu_C^2 + \sigma_{CC}\right] - k_2\mu_C
\end{aligned} \tag{5.15}$$

the fluctuation dynamics matrix $\mathbf{M}$ as

$$\begin{aligned}
\mathbf{M} &= \mathbf{S}_I\mathbf{K}\left[\mathbf{C}^T + 2\mathcal{D}^T(\mathbf{I}_1 \otimes \mu)\right] \\
&= \overbrace{\begin{bmatrix} k_1 & -k_2 \end{bmatrix}}^{\mathbf{S}_I\mathbf{K}} \overbrace{\begin{bmatrix} -(\pi_1 + \pi_2) + 2\mu_C \\ 1 \end{bmatrix}}^{\mathbf{C}^T + 2\mathcal{D}^T(\mathbf{I}_1 \otimes \mu)} \\
&= k_1\left[2\mu_C - (\pi_1 + \pi_2)\right] - k_2
\end{aligned} \tag{5.16}$$

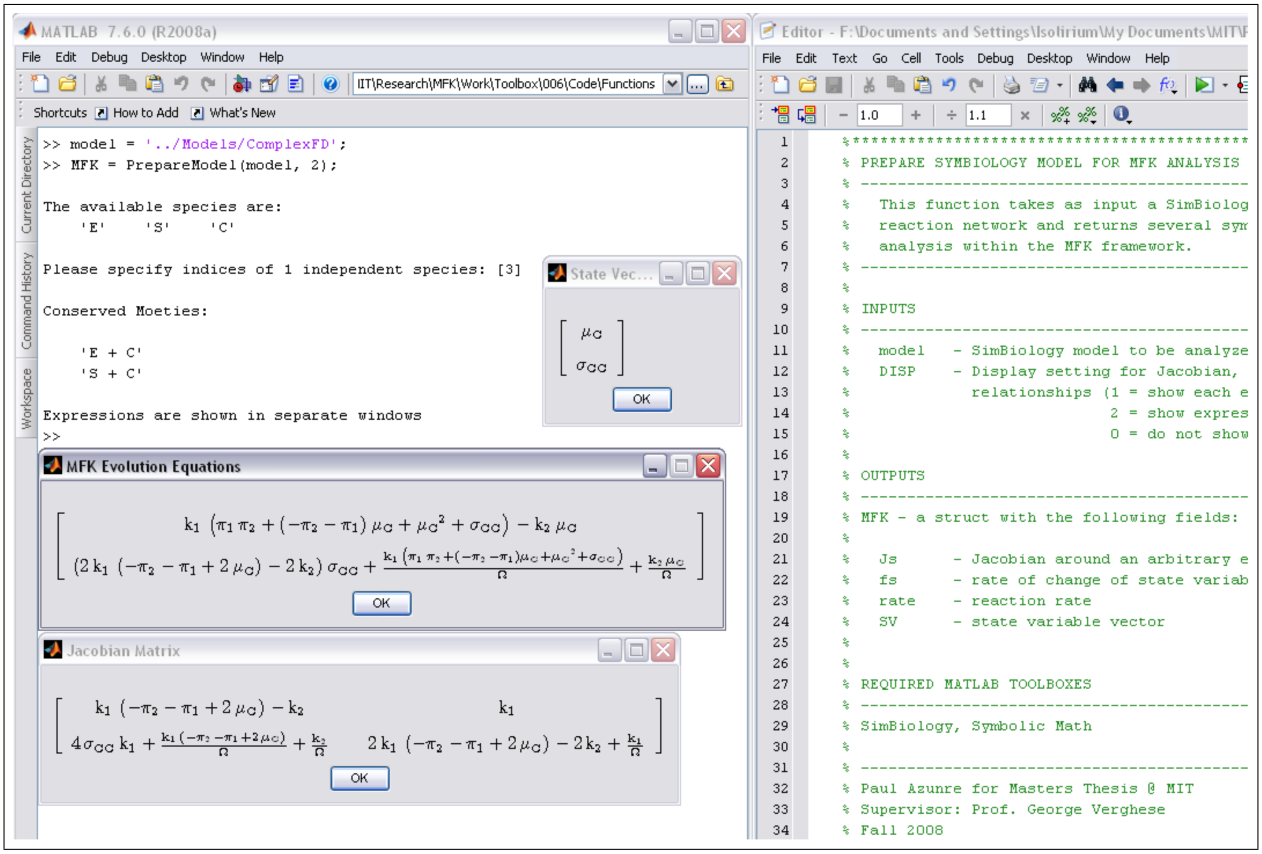
and the covariance evolution equation as

$$\begin{aligned}
\frac{d\sigma_{CC}}{dt} &= E_1(\mathbf{I}_1 \otimes \mathbf{M} + \mathbf{M} \otimes \mathbf{I}_1)F_1 vech\{\mathbf{V}\} + \frac{1}{\Omega}E_1(\mathbf{S}_I * \mathbf{S}_I)\mathbf{r} \\
&= 2\mathbf{M}\sigma_{CC} + \frac{1}{\Omega}\overbrace{\begin{bmatrix} 1 & 1 \end{bmatrix}}^{E_1(\mathbf{S}_I * \mathbf{S}_I)} \overbrace{\begin{bmatrix} k_1\pi_1\pi_2 - (\pi_1 + \pi_2)\,\mu_C + \mu_C^2 + \sigma_{CC} \\ k_2\mu_C \end{bmatrix}}^{\mathbf{r}} \\
&= 2\sigma_{CC}\left(k_1\left[2\mu_C - (\pi_1 + \pi_2)\right] - k_2\right) \\
&\quad + \frac{1}{\Omega}\left(k_1\left[\pi_1\pi_2 - (\pi_1 + \pi_2)\,\mu_C + \mu_C^2 + \sigma_{CC}\right] + k_2\mu_C\right)
\end{aligned} \tag{5.17}$$

The MFK Jacobian matrix is obtained as

$$\begin{aligned}
\mathbf{J} &= \begin{bmatrix} \frac{d}{d\mu_C}\left\{\frac{d\mu_C}{dt}\right\} & \frac{d}{d\sigma_{cc}}\left\{\frac{d\mu_C}{dt}\right\} \\ \frac{d}{d\mu_C}\left\{\frac{d\sigma_{CC}}{dt}\right\} & \frac{d}{d\sigma_{CC}}\left\{\frac{d\sigma_{CC}}{dt}\right\} \end{bmatrix} \\
&= \begin{bmatrix} k_1\left[2\mu_C - (\pi_1 + \pi_2)\right] - k_2 & k_1 \\ 4k_1\sigma_{CC} + \frac{1}{\Omega}\left(k_1\left[2\mu_C - (\pi_1 + \pi_2)\right] + k_2\right) & 2k_1\left[2\mu_C - (\pi_1 + \pi_2)\right] - 2k_2 + \frac{k_1}{\Omega} \end{bmatrix}
\end{aligned} \tag{5.18}$$

All of the preceding analysis, which becomes increasingly tedious with increasing system complexity, is automated by the toolbox through a function call to `PrepareModel`. A screenshot of the independent species specification procedure, along with returned MFK evolution equations, the Jacobian matrix and the conservation relations, is shown in Figure 5-1. These match the derived expressions exactly. Note a section of the documentation for function `PrepareModel` on the right (found at the beginning of the function file and containing specific instructions of use).


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
IIT\Research\MFK\Work\Toolbox\006\Code\Functions
Shortcuts How to Add What's New
Current Directory
Command History
Workspace
State Vec...
OK
MFK Evolution Equations
OK
Jacobian Matrix
OK
Editor - F:\Documents and Settings\Isolirium\My Documents\MIT\F
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1

```
>> model = '../Models/ComplexFD';
>> MFK = PrepareModel(model, 2);

The available species are:
    'E'    'S'    'C'

Please specify indices of 1 independent species: [3]

Conserved Moeties:

    'E + C'
    'S + C'

Expressions are shown in separate windows
>>
```

$$\begin{bmatrix} \mu_C \\ \sigma_{CC} \end{bmatrix}$$

$$\begin{bmatrix} k_1\ (\pi_1\,\pi_2 + (-\pi_2 - \pi_1)\,\mu_C + {\mu_C}^2 + \sigma_{CC}) - k_2\,\mu_C \\ (2\,k_1\ (-\pi_2 - \pi_1 + 2\,\mu_C) - 2\,k_2)\,\sigma_{CC} + \frac{k_1\,(\pi_1\,\pi_2 + (-\pi_2 - \pi_1)\mu_C + {\mu_C}^2 + \sigma_{CC})}{\Omega} + \frac{k_2\,\mu_C}{\Omega} \end{bmatrix}$$

$$\begin{bmatrix} k_1\ (-\pi_2 - \pi_1 + 2\,\mu_C) - k_2 & k_1 \\ 4\,\sigma_{CC}\,k_1 + \frac{k_1\,(-\pi_2 - \pi_1 + 2\,\mu_C)}{\Omega} + \frac{k_2}{\Omega} & 2\,k_1\ (-\pi_2 - \pi_1 + 2\,\mu_C) - 2\,k_2 + \frac{k_1}{\Omega} \end{bmatrix}$$

```
%*******************************************
% PREPARE SYMBIOLOGY MODEL FOR MFK ANALYSIS
% -----------------------------------------
%   This function takes as input a SimBiolog
%   reaction network and returns several sym
%   analysis within the MFK framework.
% -----------------------------------------
%
% INPUTS
% -----------------------------------------
%   model   - SimBiology model to be analyze
%   DISP    - Display setting for Jacobian,
%             relationships (1 = show each e
%                            2 = show expres
%                            0 = do not show
%
% OUTPUTS
% -----------------------------------------
% MFK - a struct with the following fields:
%
%   Js      - Jacobian around an arbitrary e
%   fs      - rate of change of state variab
%   rate    - reaction rate
%   SV      - state variable vector
%
%
% REQUIRED MATLAB TOOLBOXES
% -----------------------------------------
% SimBiology, Symbolic Math
%
% -----------------------------------------
% Paul Azunre for Masters Thesis @ MIT
% Supervisor: Prof. George Verghese
% Fall 2008
```

Figure 5-1: *[Color] PrepareModel function call and output for CFD*

Computational analysis

In the thermodynamic limit (Equation (1.13)) the MFK evolution equations reduce to

$$\left[\frac{dz}{dt}\right]_{\mho=0} = \begin{bmatrix} \frac{d\mu_C}{dt} \\ \frac{d\sigma_{CC}}{dt} \end{bmatrix}_{\mho=0} = \begin{bmatrix} k_1\left[\pi_1\pi_2 - (\pi_1+\pi_2)\,\mu_C + \mu_C^2 + \sigma_{CC}\right] - k_2\mu_C \\ 2\sigma_{CC}\left(k_1\left[2\mu_C - (\pi_1+\pi_2)\right] - k_2\right) \end{bmatrix} \tag{5.19}$$

and the Jacobian reduces to

$$[\mathbf{J}]_{\mho=0} = \begin{bmatrix} k_1\left[2\mu_C - (\pi_1+\pi_2)\right] - k_2 & k_1 \\ 4k_1\sigma_{CC} & 2k_1\left[2\mu_C - (\pi_1+\pi_2)\right] - 2k_2 \end{bmatrix} \tag{5.20}$$

If σ_{CC} is initialized to be positive, and noting from Equations (5.6) and (5.7) that

$$2\mu_C - (\pi_1 + \pi_2) \leq 0$$

we conclude that

$$\frac{d\sigma_{CC}}{dt} < 0$$

Thus, σ_{CC} must approach 0 with time. Also, since

$$\left[\frac{d\sigma_{CC}}{dt}\right]_{(\mho,\sigma_{CC})=(0,0)} = 0 \tag{5.21}$$

$\sigma_{CC} = 0$ is an equilibrium covariance value (regardless of the equilibrium value of μ) in the thermodynamic limit (we have already seen this specified as the MFK setting $\{\Omega, \mathbf{V}\} = \{\infty, \mathbf{0}\}$). The assumption that initial concentrations are known exactly allows setting initial covariances to 0 and the mean evolution to capture MAK dynamics.

In what follows, the initial concentration vector is set to

$$y(0) = \begin{bmatrix} y_E(0) \\ y_S(0) \\ y_C(0) \end{bmatrix} = \begin{bmatrix} 15 \\ 15 \\ 15 \end{bmatrix} \tag{5.22}$$

the reaction rate constants are all set to 1, so that

$$\mathbf{K} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}_2 \tag{5.23}$$

and the concentrations of conserved moieties are

$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \begin{bmatrix} y_C + y_E \\ y_C + y_S \end{bmatrix} = \begin{bmatrix} 30 \\ 30 \end{bmatrix} \tag{5.24}$$

Following the preceding discussion, the MFK state vector is initialized as

$$z\,(0) = \begin{bmatrix} \mu_C(0) \\ \sigma_{CC}(0) \end{bmatrix} = \begin{bmatrix} 15 \\ 0 \end{bmatrix} \tag{5.25}$$

Setting the mean evolution equation to 0 with $\sigma_{CC} = 0$ yields two possible MAK equilibrium mean concentrations of 25 and 36. However, 36 is not realizable since $\mu_C \leq 30$. Hence the MFK equilibrium corresponding to MAK is

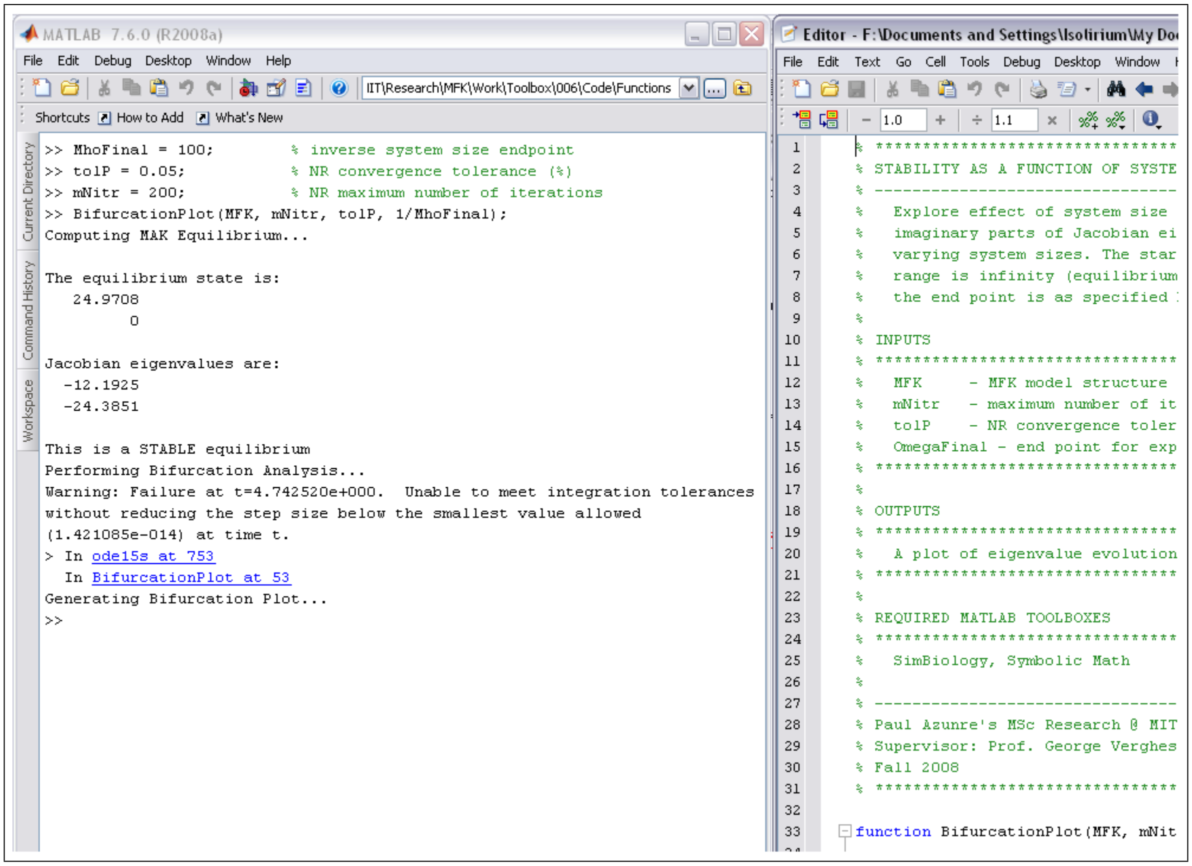
$$\bar{z}_{MAK} = \begin{bmatrix} 25 \\ 0 \end{bmatrix} \tag{5.26}$$

The Jacobian at this point is

$$\mathbf{J}_{MAK} = [\mathbf{J}]_{(\mho,z)=(0,\bar{z}_{MAK})} = \begin{bmatrix} -11 & 1 \\ 0 & -22 \end{bmatrix}$$

This is an upper triangular matrix with eigenvalues equal to its diagonal elements, which are both real and negative. Thus, this is a locally stable equilibrium. Exploring its variation in value and stability with $\mho$ provides an insight on the connection between MAK and MFK. Assuming that the system is stable (stability may be ascertained from the behavior of an exact SSA trajectory) and that this is the realized equilibrium (which happens to be the case for this simple system, as we shall soon see), the range of $\mho$ where this equilibrium is stable indicates a regime where MFK is a useful approximation for system behavior. This is the essence of the MAK equilibrium local bifurcation analysis.

The screenshot in Figure 5-2 shows a call to function `BifurcationPlot`, along with function text output, which confirms the MAK equilibrium and corresponding Jacobian eigenvalues that have been computed. Note that the slight numeric deviation of function output from calculated values is a consequence of the nonexactness of the Newton-Raphson solution due to nonrestrictive convergence tolerance (0.05 only).


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
IIT\Research\MFK\Work\Toolbox\006\Code\Functions
Shortcuts How to Add What's New
Current Directory
Command History
Workspace
Editor - F:\Documents and Settings\Isolirium\My Do
File Edit Text Go Cell Tools Debug Desktop Window
1.0
1.1

```
>> MhoFinal = 100;        % inverse system size endpoint
>> tolP = 0.05;           % NR convergence tolerance (%)
>> mNitr = 200;           % NR maximum number of iterations
>> BifurcationPlot(MFK, mNitr, tolP, 1/MhoFinal);
Computing MAK Equilibrium...

The equilibrium state is:
   24.9708
         0

Jacobian eigenvalues are:
  -12.1925
  -24.3851

This is a STABLE equilibrium
Performing Bifurcation Analysis...
Warning: Failure at t=4.742520e+000.  Unable to meet integration tolerances
without reducing the step size below the smallest value allowed
(1.421085e-014) at time t.
> In ode15s at 753
  In BifurcationPlot at 53
Generating Bifurcation Plot...
>>
```

```
% ********************************
% STABILITY AS A FUNCTION OF SYSTE
% --------------------------------
%   Explore effect of system size
%   imaginary parts of Jacobian ei
%   varying system sizes. The star
%   range is infinity (equilibrium
%   the end point is as specified
%
% INPUTS
% ********************************
%   MFK     - MFK model structure
%   mNitr   - maximum number of it
%   tolP    - NR convergence toler
%   OmegaFinal - end point for exp
% ********************************
%
% OUTPUTS
% ********************************
%   A plot of eigenvalue evolution
% ********************************
%
% REQUIRED MATLAB TOOLBOXES
% ********************************
%   SimBiology, Symbolic Math
%
% --------------------------------
% Paul Azunre's MSc Research @ MIT
% Supervisor: Prof. George Verghes
% Fall 2008
% ********************************

function BifurcationPlot(MFK, mNit
```

Figure 5-2: *[Color] BifurcationPlot function call and text output for CFD*

The local bifurcation plot produced by the `BifurcationPlot` function call in Figure 5-2 is shown in Figure 5-3. It is observed that the equilibrium becomes unstable, evidenced by the real part of a Jacobian eigenvalue becoming positive, at $\mho \approx 4.7$. Since the imaginary parts of both eigenvalues remain zero throughout the analysis, the said eigenvalue can be visualized as crossing the $j\omega$-axis through the origin of the complex plane. At the crossing point, the Jacobian becomes singular and local bifurcation analysis cannot proceed further. This causes the integration tolerance warning shown in Figure 5-2.

To picture the bifurcation in the time-domain, three trajectories are generated in stable, (approximately) marginally stable and unstable MFK regimes ($\mho = 1, 4.5, 5$ respectively) with `TrajectoryPlot`. These trajectories are shown in Figures 5-4, 5-5 and

5-6. Real system behavior (as captured by the SSA trajectories) is observed to become more discrete and stochastic in nature but remain stable with increasing $\mho$. Thus, MFK is only a useful approximation in the range $\mho \in [0, 4.7]$. The MFK trajectories capture increasing randomness via the growth of the variance error bounds. The fact that there is no way for the MAK model to capture this effect provides intuition on why it becomes inaccurate at smaller volumes. However, it appears that the continuous MFK equations can only capture noisy behavior up to some variance threshold, beyond which they collectively become unstable.

To study evolution of local coupling between means and covariances, a participation factor plot of μ_C in each of the two modes $\{\lambda_i\}_{i=1}^{2}$ along the equilibrium evolution path is generated by function `ParticipationFactorPlot`. This plot is shown in Figure 5-7. Since μ_C participates only in λ_1 at $\mho = 0$ (and σ_{CC} participates only in λ_2) μ_C and σ_{CC} are completely decoupled here. Hence, the evolution of μ_C can be completely described by itself alone and the inclusion of σ_{CC} in the model (i.e., the use of MFK) is not needed. The coupling is observed to become steadily stronger with increasing $\mho$, but to remain relatively weak until $\mho \approx 1$. Hence, MAK is expected to be an accurate reduced model for system behavior up to $\mho \approx 1$. Beyond this point, the significant coupling between μ_C and σ_{CC} indicates that the use of MFK would provide more accurate results.

5.3.2 Irreversible enzyme reaction system (IERS)

Having confirmed toolbox output credibility analytically for CFD, the analytic results for IERS are not derived explicitly here. Key parameter values that were used in simulations are listed here for reproducibility and concreteness. Analysis of local dynamics is also presented, just as for CFD.

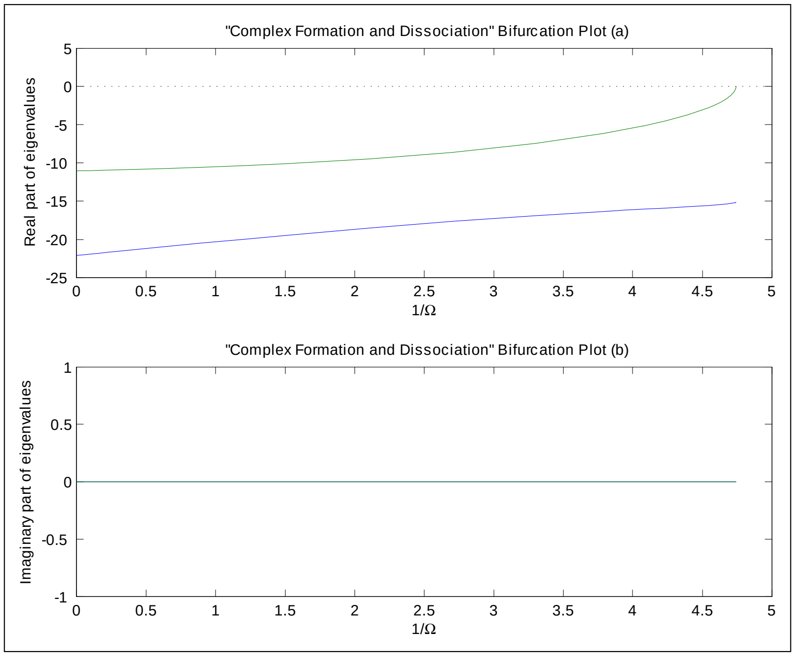


Figure 5-3: *[Color] Local bifurcation plot for CFD*

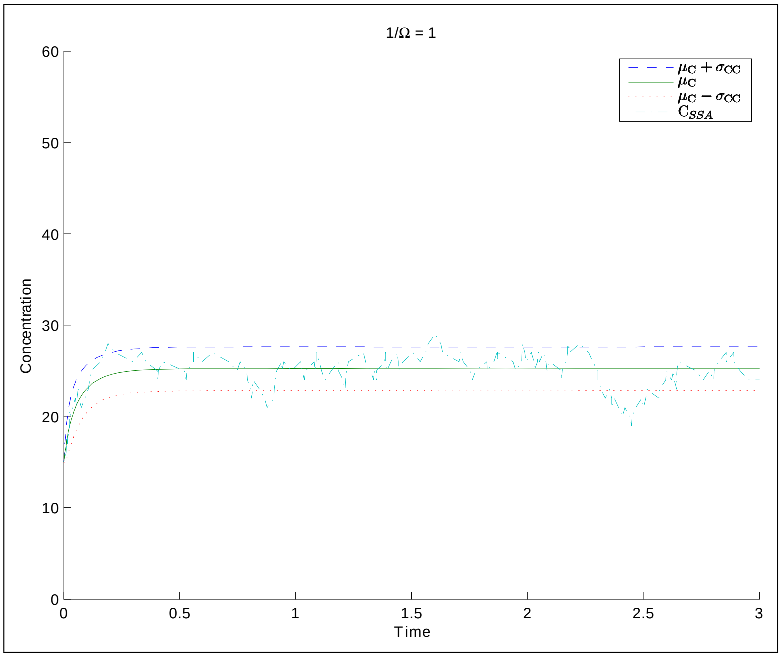


Figure 5-4: *[Color] CFD trajectories in stable MFK regime*

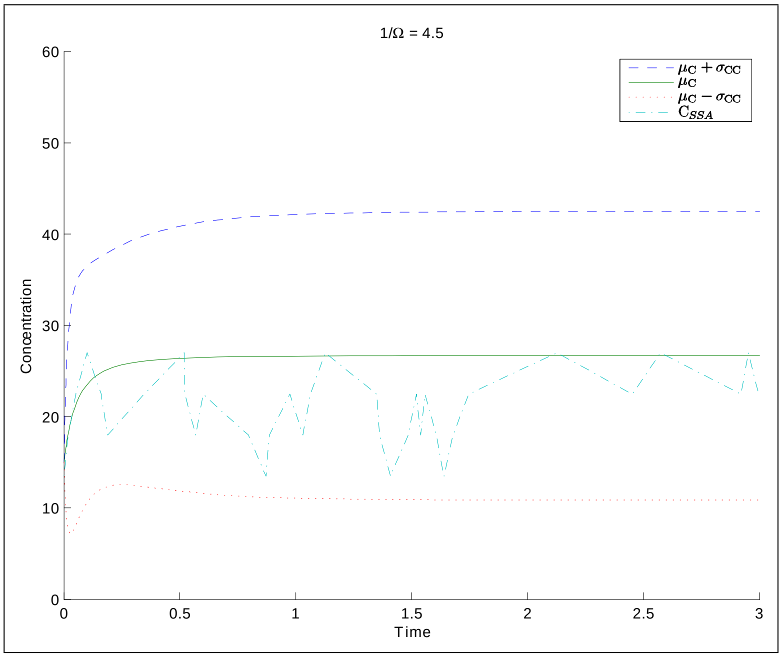


Figure 5-5: *[Color] CFD trajectories in (approximately) marginally stable MFK regime*

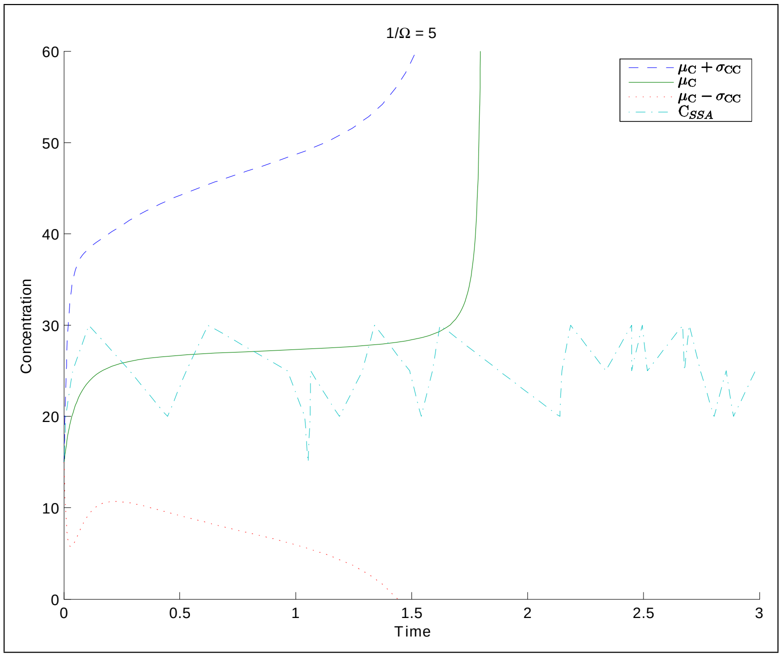


Figure 5-6: *[Color] CFD trajectories in unstable MFK regime*

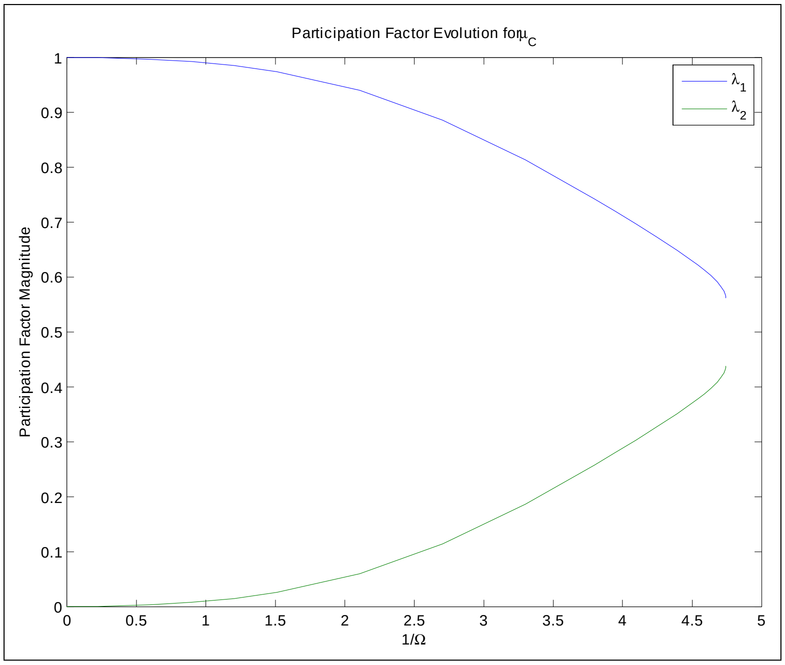


Figure 5-7: *[Color] Participation factor evolution for CFD*

Preliminaries

The stoichiometry vectors for reactions R_1, R_2 and R_3 (s_1, s_2 and s_3 respectively), are grouped into the stoichiometry matrix $\mathbf{S}$ as

$$\mathbf{S} = \left[\begin{array}{ccc} s_1 & s_2 & s_3 \end{array}\right] = \begin{array}{c|ccc|} & R_1 & R_2 & R_3 \\ E & -1 & 1 & 1 \\ S & -1 & 1 & 0 \\ C & 1 & -1 & -1 \\ P & 0 & 0 & 1 \end{array} \tag{5.27}$$

The correspondence of rows to species and columns to reactions has been made explicit. $\mathbf{S}$ corresponds to the species concentration vector

$$y = \begin{bmatrix} y_E \\ y_S \\ y_C \\ y_P \end{bmatrix} \tag{5.28}$$

Analytic expressions

A function call to `PrepareModel` automates moiety conservation analysis and obtains the MFK evolution equations and the Jacobian. A screenshot of the function call and most of its output is shown in Figure 5-8. This is an example where each expression (each MFK evolution equation and each row of the Jacobian) must be shown in a separate window to be visible (due to the prohibitive lengths of expressions). The screenshot exposes the complexity of the underlying MFK model and makes one appreciate the symbolic capabilities of the prototype.

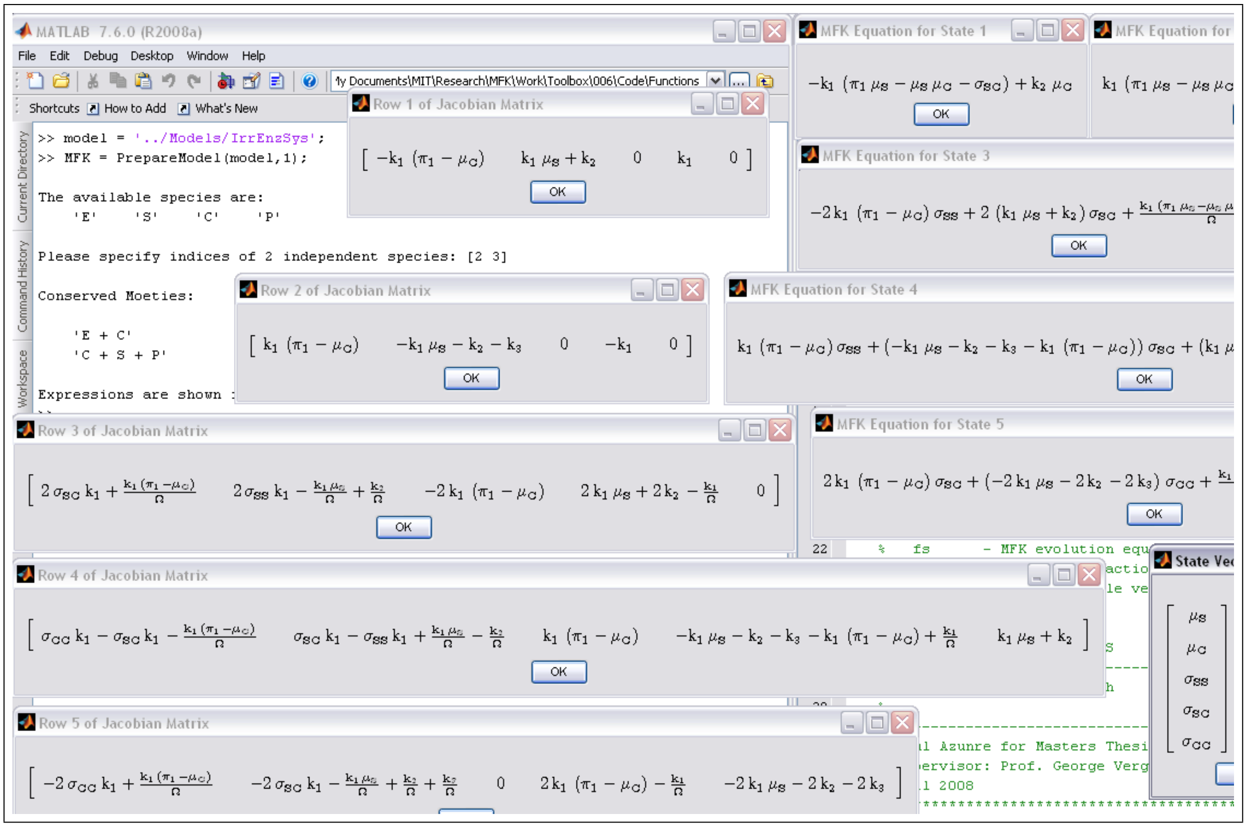
Computational analysis

In what follows, the initial species concentration vector is set to

$$y(0) = \begin{bmatrix} y_E(0) \\ y_S(0) \\ y_C(0) \\ y_P(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 0 \\ 0 \end{bmatrix} \tag{5.29}$$

the reaction rate constants are all set to 1, so that

$$\mathbf{K} = \mathbf{I}_3 \tag{5.30}$$



```
>> model = '../Models/IrrEnzSys';
>> MFK = PrepareModel(model,1);

The available species are:
    'E'    'S'    'C'    'P'

Please specify indices of 2 independent species: [2 3]

Conserved Moeties:

    'E + C'
    'C + S + P'

Expressions are shown :
```

$[\ -k_1\ (\pi_1-\mu_C) \quad k_1\,\mu_S+k_2 \quad 0 \quad k_1 \quad 0\ ]$

$[\ k_1\ (\pi_1-\mu_C) \quad -k_1\,\mu_S-k_2-k_3 \quad 0 \quad -k_1 \quad 0\ ]$

$[\ 2\,\sigma_{SC}\,k_1+\frac{k_1\,(\pi_1-\mu_C)}{\Omega} \quad 2\,\sigma_{SS}\,k_1-\frac{k_1\,\mu_S}{\Omega}+\frac{k_2}{\Omega} \quad -2\,k_1\ (\pi_1-\mu_C) \quad 2\,k_1\,\mu_S+2\,k_2-\frac{k_1}{\Omega} \quad 0\ ]$

$[\ \sigma_{CC}\,k_1-\sigma_{SC}\,k_1-\frac{k_1\,(\pi_1-\mu_C)}{\Omega} \quad \sigma_{SC}\,k_1-\sigma_{SS}\,k_1+\frac{k_1\,\mu_S}{\Omega}-\frac{k_2}{\Omega} \quad k_1\ (\pi_1-\mu_C) \quad -k_1\,\mu_S-k_2-k_3-k_1\ (\pi_1-\mu_C)+\frac{k_1}{\Omega} \quad k_1\,\mu_S+k_2\ ]$

$[\ -2\,\sigma_{CC}\,k_1+\frac{k_1\,(\pi_1-\mu_C)}{\Omega} \quad -2\,\sigma_{SC}\,k_1-\frac{k_1\,\mu_S}{\Omega}+\frac{k_2}{\Omega}+\frac{k_3}{\Omega} \quad 0 \quad 2\,k_1\ (\pi_1-\mu_C)-\frac{k_1}{\Omega} \quad -2\,k_1\,\mu_S-2\,k_2-2\,k_3\ ]$

$-k_1\ (\pi_1\,\mu_S-\mu_S\,\mu_C-\sigma_{SC})+k_2\,\mu_C$

$k_1\ (\pi_1\,\mu_S-\mu_S\,\mu_C$

$-2\,k_1\ (\pi_1-\mu_C)\,\sigma_{SS}+2\ (k_1\,\mu_S+k_2)\,\sigma_{SC}+\frac{k_1\,(\pi_1\,\mu_S-\mu_S\,\mu}{\Omega}$

$k_1\ (\pi_1-\mu_C)\,\sigma_{SS}+(-k_1\,\mu_S-k_2-k_3-k_1\ (\pi_1-\mu_C))\,\sigma_{SC}+(k_1\,\mu$

$2\,k_1\ (\pi_1-\mu_C)\,\sigma_{SC}+(-2\,k_1\,\mu_S-2\,k_2\ -2\,k_3)\,\sigma_{CC}+\frac{k_1}{}$

```
22    %    fs        - MFK evolution equ
                                  actio
                                  le ve

                                  S
                                  -----
                                  h

   --------------------------------
   l Azunre for Masters Thesi
   ervisor: Prof. George Verg
   l 2008
   ***********************************
```

$$\begin{bmatrix} \mu_S \\ \mu_C \\ \sigma_{SS} \\ \sigma_{SC} \\ \sigma_{CC} \end{bmatrix}$$

Figure 5-8: *[Color] PrepareModel function call and output for IERS*

and the concentrations of conserved moieties are

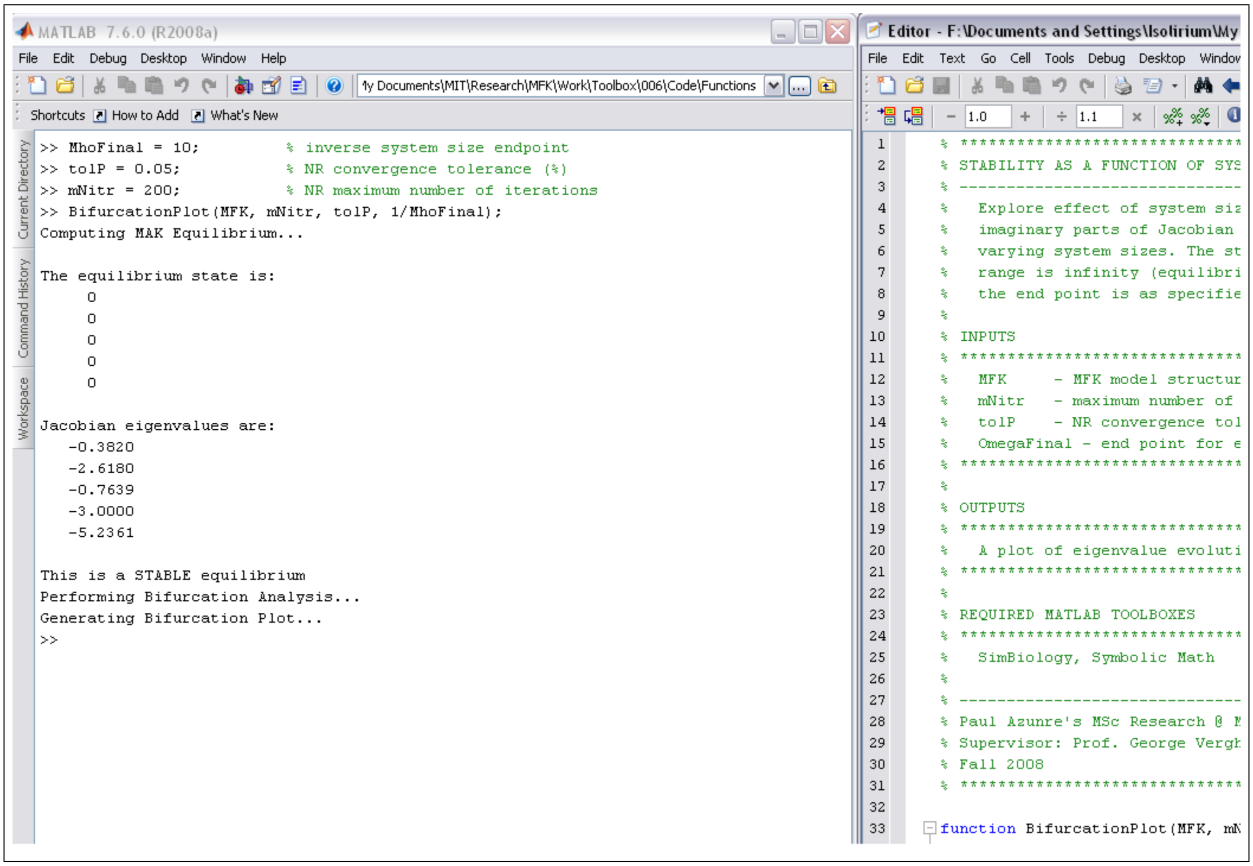
$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \begin{bmatrix} y_E + y_C \\ y_C + y_S + y_P \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix} \tag{5.31}$$

The MFK state vector is initialized as

$$z\,(0) = \begin{bmatrix} \mu_S \\ \mu_C \\ \sigma_{SS} \\ \sigma_{SC} \\ \sigma_{CC} \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.32}$$

Local bifurcation analysis is automated by a `BifurcationPlot` function call in Figure

5-9. The MAK equilibrium is the origin and its bifurcation plot is shown in Figure 5-10. It is observed that the equilibrium becomes unstable at $\mho \approx 4.5$. The eigenvalue which crosses the $j\omega$-axis does so away from the origin of the complex plane (the imaginary part of the eigenvalue is nonzero at the crossing point). Hence, the Jacobian does not become singular and local bifurcation analysis proceeds without any warnings throughout the $\mho$ range of interest.


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
ly Documents\MIT\Research\MFK\Work\Toolbox\006\Code\Functions
Shortcuts How to Add What's New
Current Directory
Command History
Workspace
Editor - F:\Documents and Settings\Isolirium\My
File Edit Text Go Cell Tools Debug Desktop Window
1.0
1.1

```
>> MhoFinal = 10;          % inverse system size endpoint
>> tolP = 0.05;            % NR convergence tolerance (%)
>> mNitr = 200;            % NR maximum number of iterations
>> BifurcationPlot(MFK, mNitr, tolP, 1/MhoFinal);
Computing MAK Equilibrium...

The equilibrium state is:
     0
     0
     0
     0
     0


Jacobian eigenvalues are:
   -0.3820
   -2.6180
   -0.7639
   -3.0000
   -5.2361

This is a STABLE equilibrium
Performing Bifurcation Analysis...
Generating Bifurcation Plot...
>>
```

```
% ******************************
% STABILITY AS A FUNCTION OF SYS
% ------------------------------
%   Explore effect of system siz
%   imaginary parts of Jacobian
%   varying system sizes. The st
%   range is infinity (equilibri
%   the end point is as specifie
%
% INPUTS
% ******************************
%   MFK      - MFK model structur
%   mNitr    - maximum number of
%   tolP     - NR convergence tol
%   OmegaFinal - end point for e
% ******************************
%
% OUTPUTS
% ******************************
%   A plot of eigenvalue evoluti
% ******************************
%
% REQUIRED MATLAB TOOLBOXES
% ******************************
%   SimBiology, Symbolic Math
%
% ------------------------------
% Paul Azunre's MSc Research @ M
% Supervisor: Prof. George Vergh
% Fall 2008
% ******************************

function BifurcationPlot(MFK, mN
```

Figure 5-9: *[Color] BifurcationPlot function call and text output for IERS*

Three trajectories are generated in stable, (approximately) marginally stable and unstable regimes ($\mho = 1, 4, 6$ respectively) with the aid of function `TrajectoryPlot`. The three trajectories are shown in Figures 5-11, 5-12 and 5-13. This example illustrates an important point. At $\mho = 1$, a valid SSA trajectory is generated while at $\mho = 4$, the SSA trajectory remains flat and never changes. The reason for this is the fractional

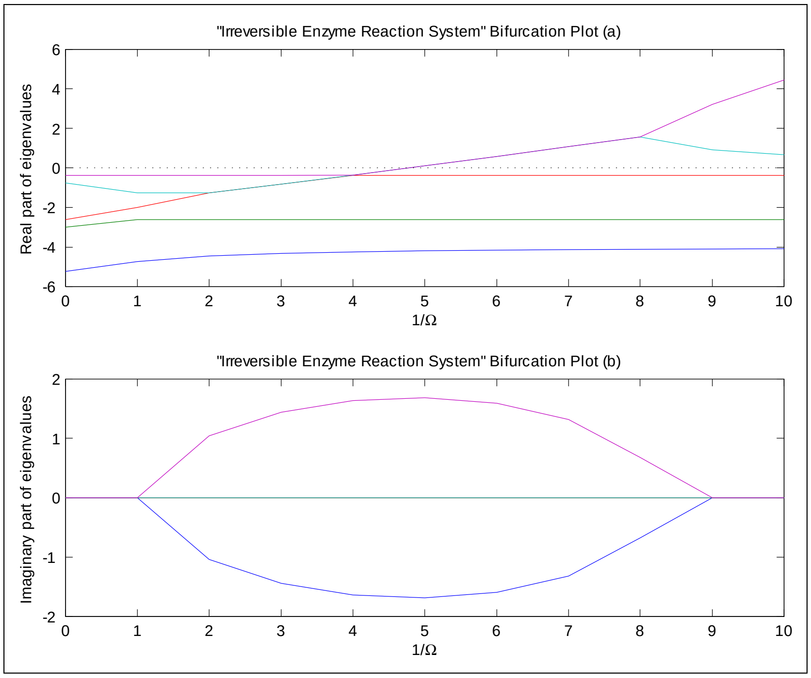


Figure 5-10: *[Color] Local bifurcation plot for IERS*

number of initial enzyme molecules at $\mho = 4$. For $y(0)$ as specified in Equation (5.29), $\mho > 1 \Rightarrow x_E(0) = \frac{y_E(0)}{\mho} < 1$. Thus, no whole enzyme molecules are initially present for the reaction to occur and this is a nonphysical regime. This highlights that an unsuspecting user may be easily misled by MFK trajectories while exploring nonphysical regimes. Inclusion of a single SSA trajectory along with MFK as a sanity check rectifies this problem and is highly recommended. For this discussion to be complete, the nonphysical regime assertion needs to be tested more thoroughly. One way of doing this is to simulate a trajectory at $\mho = 1.1$, which just satisfies the condition of interest $\mho > 1$. This trajectory is shown in Figure 5-14. The SSA trajectory is indeed flat here, lending credibility to the assertion made. A feature that may be interesting for future exploration is the "onion" shape of MFK trajectories in Figure 5-12.

To study evolution of local coupling between means and covariances, a participation factor plot of μ_S in each of the five modes $\{\lambda_i\}_{i=1}^{5}$ along the equilibrium evolution path is generated by function `ParticipationFactorPlot`. The plot is shown in Figure 5-15. The state variable μ_S is observed to participate only in the modes λ_1 and λ_2 at $\mho = 0$

(the modes corresponding to the means). This suggests that means and covariances are completely decoupled here. The coupling appears to increase more rapidly with $\mho$ here than in the case of CFD, quickly making MFK the preferred model for capturing system behavior.

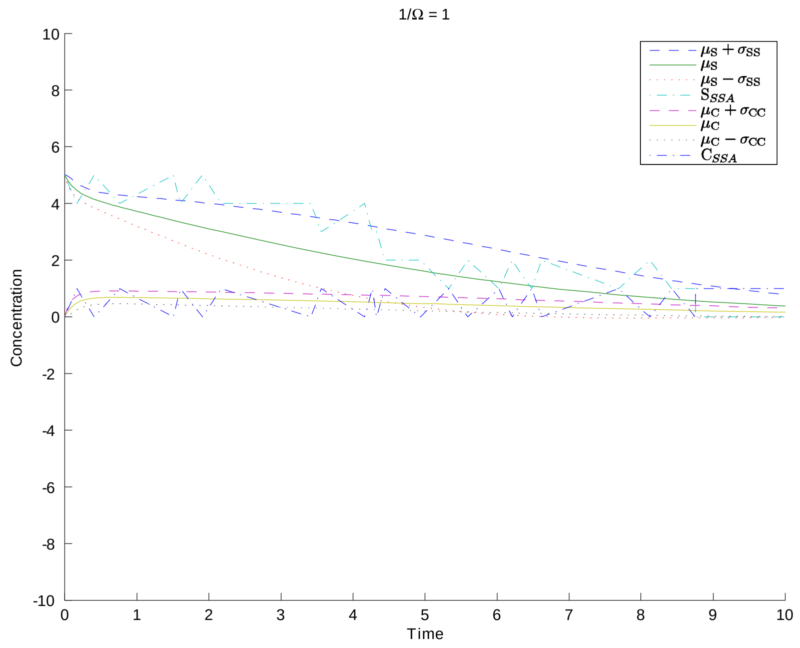


Figure 5-11: *[Color] IERS trajectories in stable MFK regime*

5.3.3 Reversible enzyme reaction system (RERS)

The discussion of results for this system will be brief. Simulation settings are listed, for completeness of discussion, as

$$S = \begin{array}{c} \\ E \\ S \\ C \\ P \end{array} \begin{array}{c} \begin{array}{cccc} R_1 & R_2 & R_3 & R_4 \end{array} \\ \begin{bmatrix} -1 & 1 & 1 & -1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \end{array} \tag{5.33}$$

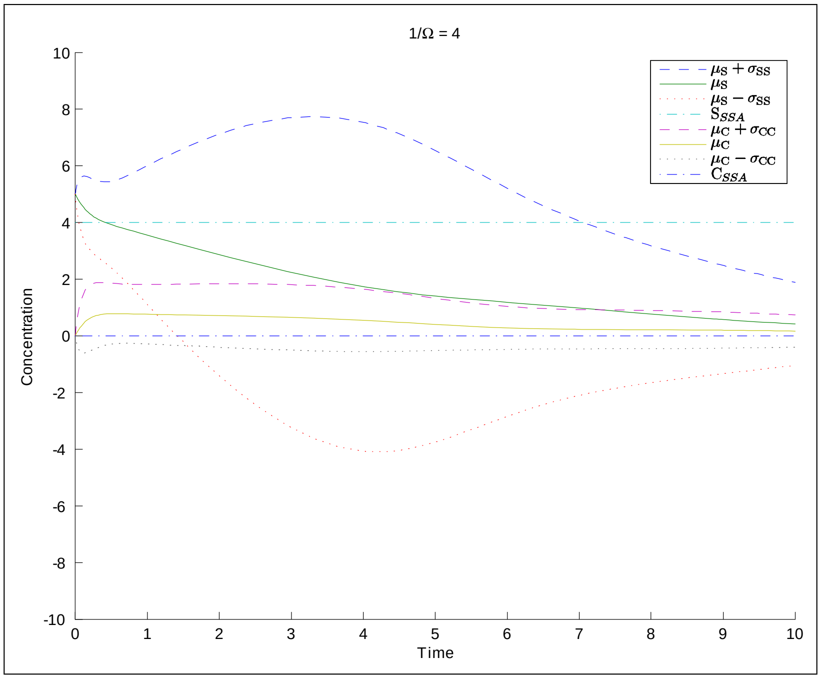


Figure 5-12: *[Color] IERS trajectories in (approximately) marginally stable MFK regime*

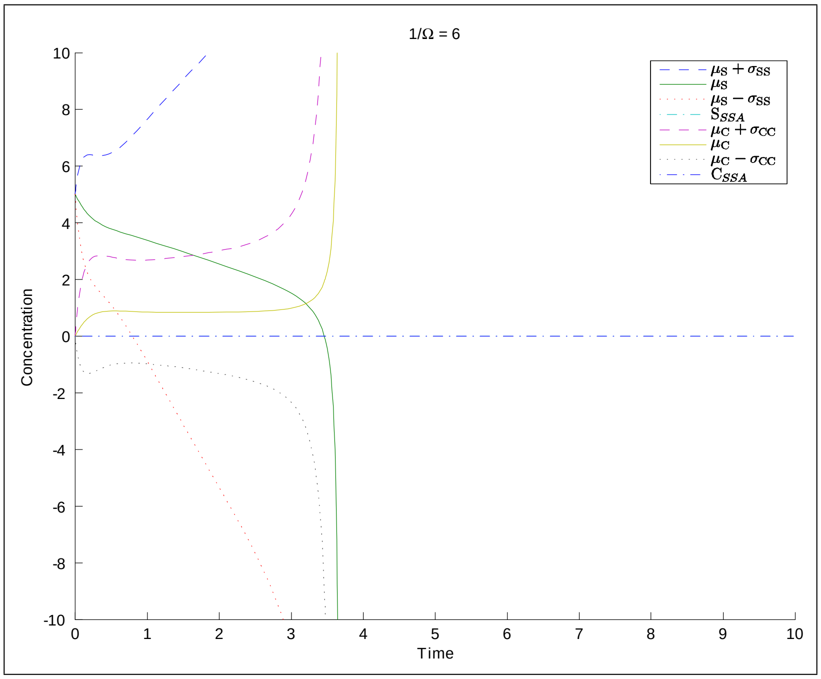


Figure 5-13: *[Color] IERS trajectories in unstable MFK regime*

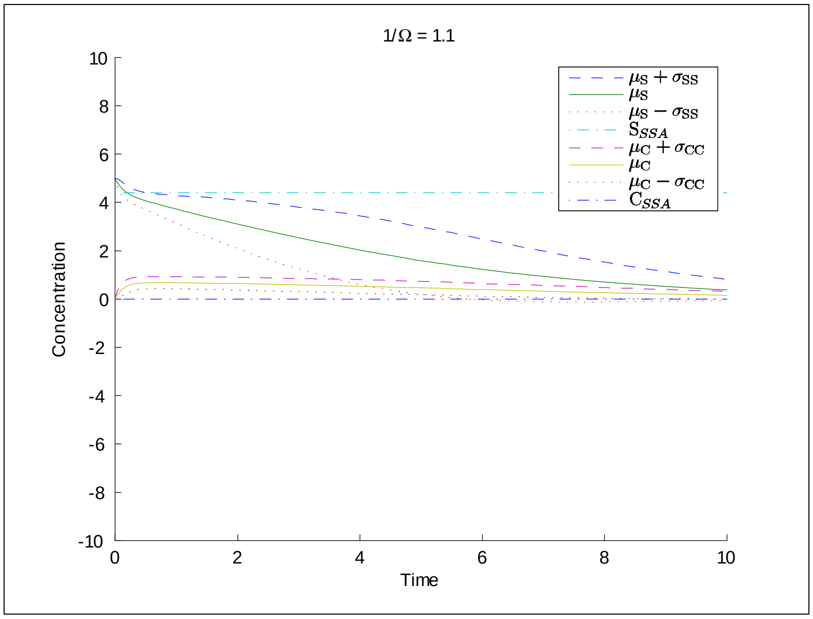


Figure 5-14: *[Color] Test of IERS SSA molecule number assertion*

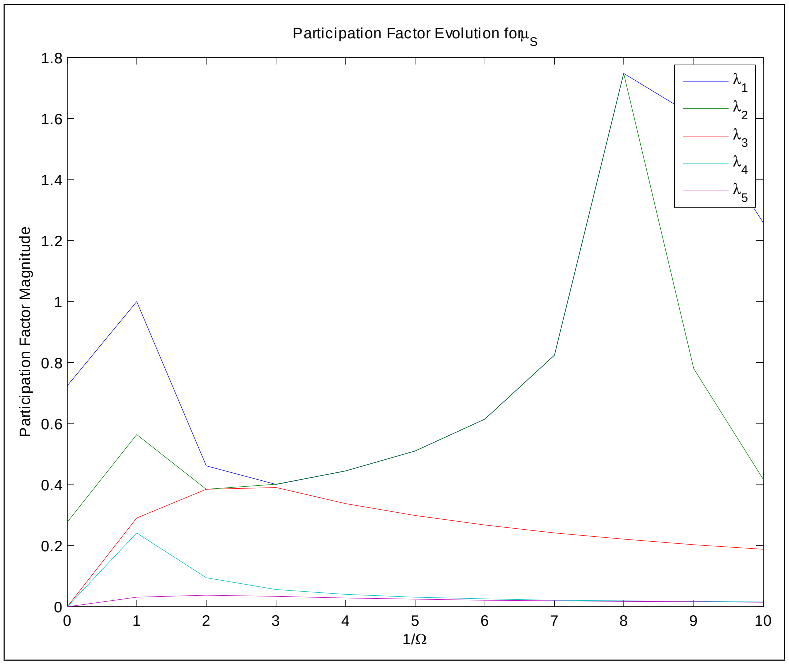


Figure 5-15: *[Color] Participation factor plot for IERS*

$$y(0) = \begin{bmatrix} y_E(0) \\ y_S(0) \\ y_C(0) \\ y_P(0) \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ 0 \\ 0 \end{bmatrix} \tag{5.34}$$

$$\pi = \begin{bmatrix} y_E + y_C \\ y_C + y_S + y_P \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \end{bmatrix} \tag{5.35}$$

$$z(0) = \begin{bmatrix} \mu_S \\ \mu_C \\ \sigma_{SS} \\ \sigma_{SC} \\ \sigma_{CC} \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.36}$$

Computational analysis

The local bifurcation plot is shown in Figure 5-16. It is observed that the Jacobian is singular at the eigenvalue crossing and that the crossing happens at $\mho \approx 4.5$. Trajectories in stable, (approximately) marginally stable and unstable MFK regimes are shown in Figures 5-17, 5-18 and 5-19. All observations regarding the randomness of SSA trajectories and its effect on the stability of MFK trajectories made in the case of CFD can be made here as well.

The participation factor plot for μ_S is shown in Figure 5-20. As is expected at this point of the discussion, means and covariances show complete decoupling at $\mho = 0$. The coupling is observed to become stronger with $\mho$, but at a much slower pace than for CFD and IERS. This suggests that MAK remains a good approximation for RERS behavior and that MFK does not become useful until $\mho \approx 2$. An opportunity to reduce the MAK model further in the range $\mho \leq 2$ may be present, due to the relatively small participation of μ_S in λ_2 as compared to λ_1.

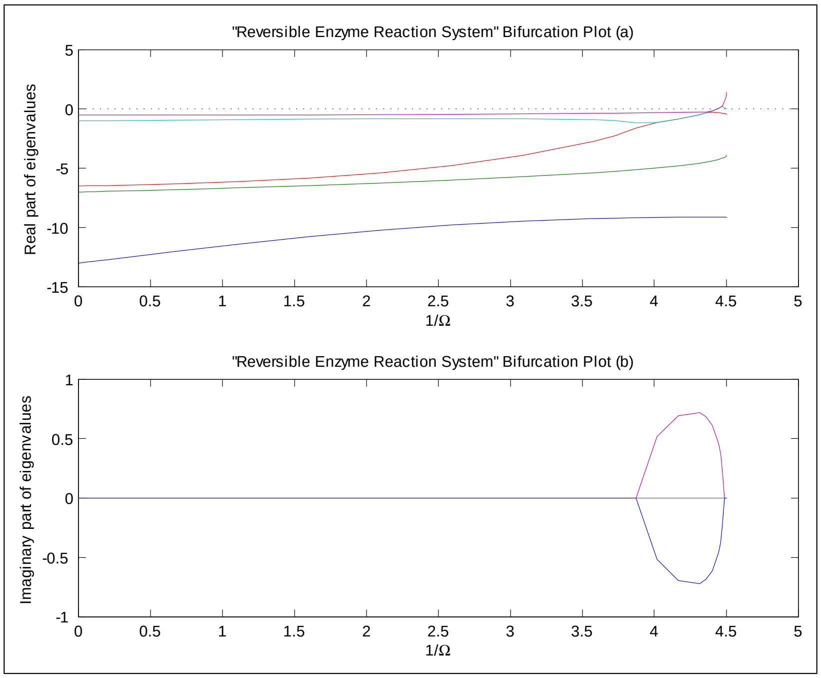


Figure 5-16: *[Color] Local bifurcation plot for RERS*

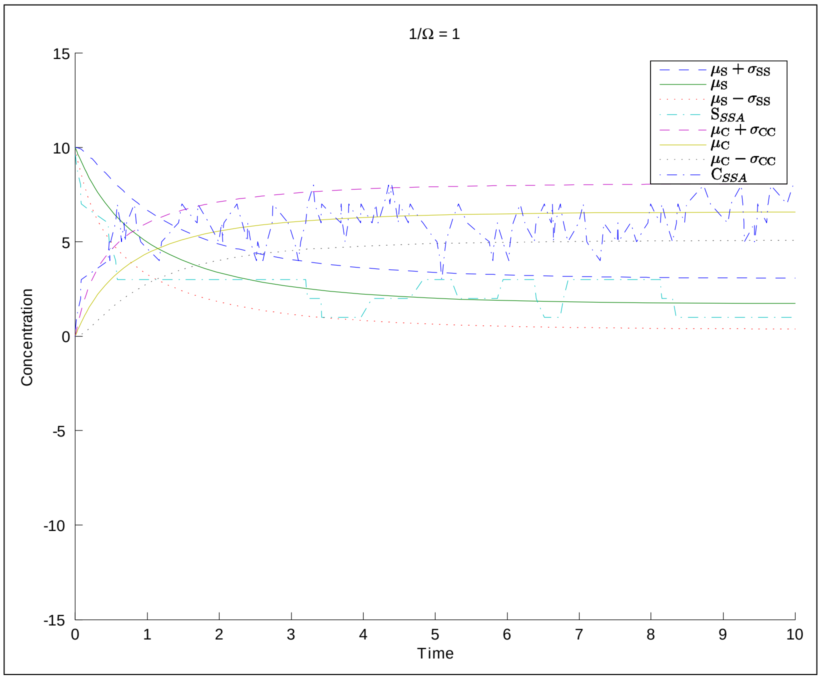


Figure 5-17: *[Color] RERS trajectories in stable MFK regime*

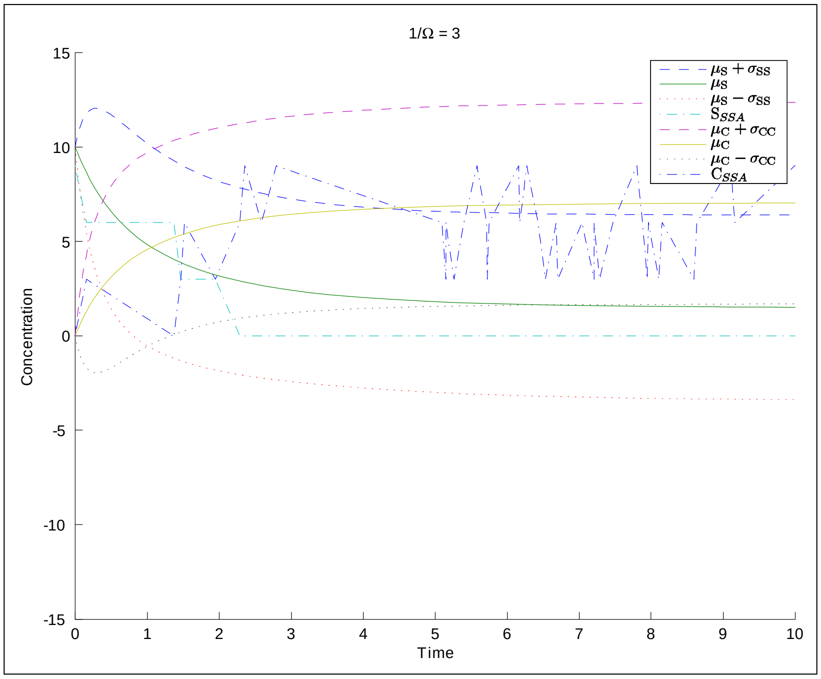


Figure 5-18: *[Color] RERS trajectories in (approximately) marginally stable MFK regime*

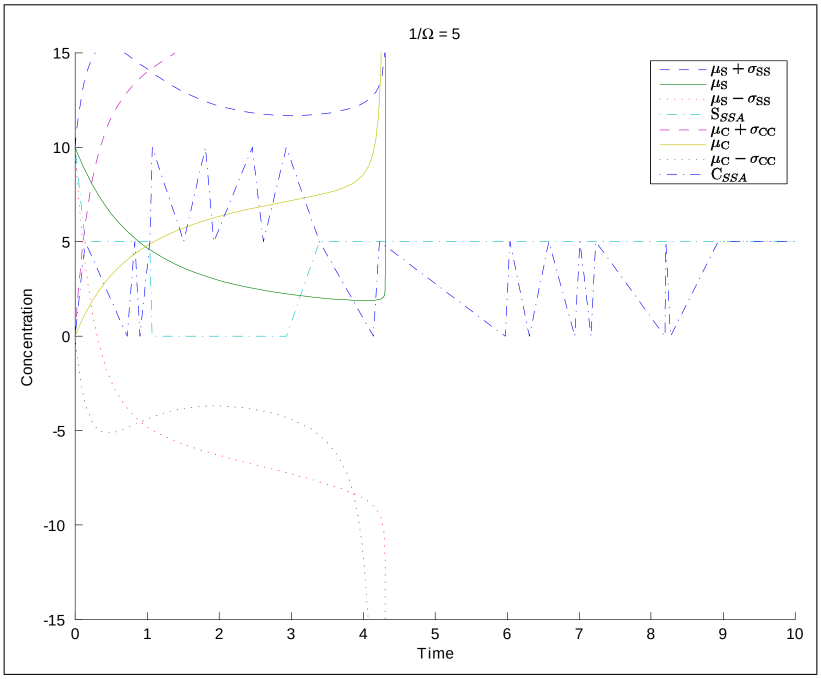


Figure 5-19: *[Color] RERS trajectories in unstable MFK regime*

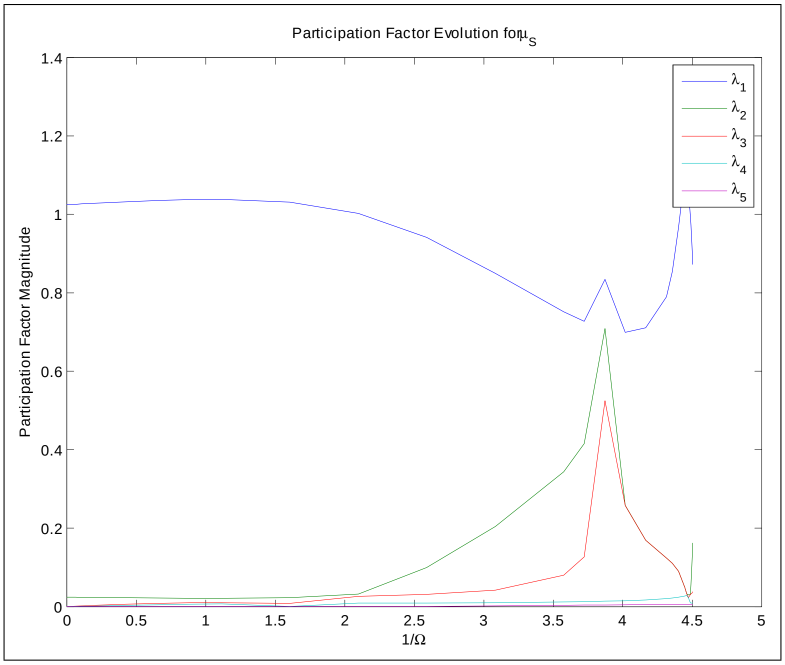


Figure 5-20: [Color] Participation factor plot for RERS

5.4 Conclusions and future work

A number of interesting observations can be made from the exploratory study performed in this chapter. Increasing randomness of system time-domain behavior with decreasing system volume, a characteristic that the MAK model cannot capture, is captured by MFK via the growth of its variance error bounds. This ability is limited to a threshold beyond which MFK becomes unstable and ceases to be a useful approximation for system behavior. This is perhaps due to third-order (or perhaps some other higher order) moments becoming more and more significant with decreasing system volume. Systematically extending the model to higher order moments via the procedure described in Chapter 2 is expected to improve its accuracy and stability.

Participation factor analysis highlights that MFK means and covariances are locally decoupled at the MAK equilibrium in the thermodynamic limit, providing an alternative explanation for why MAK is an accurate approximation for system behavior there. The ability of participation factor analysis to give a sense of ranges of system volume where switching from MAK to MFK is prudent, as well as its ability to highlight opportunities for model reduction, were noted. The danger of MFK providing misleading results in nonphysical regimes, and its alleviation via the inclusion of a single SSA trajectory, was demonstrated.

Several studies could be performed from here. Bifurcations with respect to other MFK parameters may provide further insights into MFK characteristics. A theoretical quantification of the MFK $\mho$-stability bound would be useful. In more complex (real) systems, local bifurcation analysis may make misleading suggestions about system behavior because the equilibrium under study can be reasonably expected to cease being the realized one at some point along the equilibrium evolution path. A way to detect this effect would render MFK local bifurcation analysis significantly more rigorous.

Appendix A

Matrix Theory Overview

This appendix provides a brief overview of matrix results used in the text. More specific results do not appear here but are provided within the relevant portions of the document, as needed (see also [10]).

A.1 Matrices and vectors

A matrix $A \in \mathbb{R}^{m \times n}$ is a rectangular table of m rows and n columns containing $m \times n$ elements, depicted as

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \tag{A.1}$$

Single elements are indexed from A as

$$a_{ij} = \{A\}_{ij} \tag{A.2}$$

In this document, matrices are represented by capital letters. Note that the notation $\mathbb{R}$ restricts elements of A to be real numbers. Analogously, the notation $\mathbb{Z}$ would restrict elements of A to be integers.

The transpose of $A \in \mathbb{R}^{m\times n}$ is an operation that converts rows to columns and vice versa, depicted as

$$A^T = \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{mn} \end{bmatrix} \tag{A.3}$$

The *trace* of a square matrix (a matrix with the same number of rows as columns) A is defined as the sum of its main diagonal elements, depicted as

$$Tr\left\{A\right\} = Tr\left\{\begin{bmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{nn} \end{bmatrix}\right\} = \sum_{i=1}^{n} a_{ii} \tag{A.4}$$

A symmetric matrix $D \in \mathbb{R}^{n\times n}$ is a square matrix that is equal to its own transpose.

The product of matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times q}$ is defined as

$$AB = \begin{bmatrix} \sum\limits_{i=1}^{n}\left[\{A\}_{1i}\{B\}_{i1}\right] & \cdots & \sum\limits_{i=1}^{n}\left[\{A\}_{1i}\{B\}_{iq}\right] \\ \vdots & \ddots & \vdots \\ \sum\limits_{i=1}^{n}\left[\{A\}_{mi}\{B\}_{i1}\right] & \cdots & \sum\limits_{i=1}^{n}\left[\{A\}_{mi}\{B\}_{iq}\right] \end{bmatrix} \tag{A.5}$$

Note that the definition of the matrix product requires the number of columns of A to be equal to the number of rows of B to be dimensionally compatible.

The identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is a square matrix with main diagonal elements equal to 1 and all other elements equal to zero, depicted as

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.6}$$

$\mathbf{I}_n$ is an example of a *diagonal matrix* and a *binary matrix*. The product of $\mathbf{I}_n$ with a matrix of compatible dimensions is the matrix itself. The subscript n in $\mathbf{I}_n$ makes the dimension of the identity matrix explicit, which is useful when writing software.

A column vector $b \in \mathbb{R}^m$ is a matrix containing m rows and only one column (hence m elements), depicted as

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{A.7}$$

Single elements are indexed from b as

$$b_i = \{b\}_i \tag{A.8}$$

A row vector is analogously a matrix containing only one row. In this document, vectors are represented by small letters.

A.2 Vectorization

The *vectorization* of matrix $C \in \mathbb{R}^{r \times s}$ is defined as the stacking of its columns into a single column vector, i.e.,

$$vec\{C\} = vec\left\{\begin{bmatrix} c_{:1} & \cdots & c_{:s} \end{bmatrix}\right\} = \begin{bmatrix} c_{:1} \\ \vdots \\ c_{:s} \end{bmatrix} \in \mathbb{R}^{rs} \tag{A.9}$$

Here, $c_{:i}$ is MATLAB-compatible notation for the i^{th} column of C. The following result relates $vec\left\{C^T\right\}$ to $vec\{C\}$:

$$vec\left\{C^T\right\} = \mathbf{P}_{sr} vec\{C\} \tag{A.10}$$

Here, $\mathbf{P}_{sr}$ is called the *commutation matrix*, a *permutation matrix* obtained by rearranging rows of $\mathbf{I}_{s \times r}$ in a specific order [15].

For symmetric matrices, the vectorization contains duplicate versions of all but the main diagonal elements, and is thus redundant. In this case, the *half-vectorization* operation that vectorizes only the lower triangular portion of the matrix, is used instead. It is denoted by the notation *vech*. The relationship of this operation to the vectorization for a symmetric matrix $D \in \mathbb{R}^{n \times n}$ is

$$\begin{aligned} vech\{D\} &= E_n vec\{D\} && \text{(A.11)} \\ F_n vech\{D\} &= vec\{D\} && \text{(A.12)} \end{aligned}$$

The binary matrices $E_n \in \mathbb{R}^{\frac{n(n+1)}{2} \times n^2}$ and $F_n \in \mathbb{R}^{n^2 \times \frac{n(n+1)}{2}}$ are aptly named the *selection* and *duplication* matrices respectively. The following useful vectorization relationship holds for matrices of compatible dimensions.

$$Tr\{A^T B\} = (vec\{A\})^T vec\{B\} \tag{A.13}$$

A.3 The Kronecker product

The *Kronecker product* between matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ is defined as

$$A \otimes B = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1q}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mq}B \end{bmatrix} \in \mathbb{R}^{mp \times nq} \tag{A.14}$$

For matrices of compatible dimensions, the vectorization and the Kronecker product are related by

$$vec\{ABC\} = \left(C^T \otimes A\right) vec\{B\} \tag{A.15}$$

A.4 The Khatri-Rao product

The *Khatri-Rao product*, denoted by the $*$ symbol, is defined as the columnwise Kronecker product for matrices with the same number of columns [16], i.e., for $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times n}$

$$A * B = \begin{bmatrix} a_{:1} \otimes b_{:1} & \cdots & a_{:n} \otimes b_{:n} \end{bmatrix} \in \mathbb{R}^{mp \times n} \tag{A.16}$$

For a diagonal matrix G and a matrix of compatible dimensions A, the following result holds.

$$(A \otimes A) \, vec\,\{G\} = (A * A) \, diag\,\{G\}$$

This is straightforward (although not easy) to verify via a brute-force index book-keeping argument.

A.5 Matrix derivatives

The derivative of a scalar function f with respect to vector $a \in \mathbb{R}^n$ is defined as

$$\frac{df}{da} = \left[\frac{df}{da_1} \cdots \frac{df}{da_n} \right] \in \mathbb{R}^{1 \times n} \tag{A.17}$$

The derivative of a scalar f with respect to matrix $A \in \mathbb{R}^{m \times n}$ is defined as

$$\frac{df}{dA} = \begin{bmatrix} \frac{df}{da_{11}} & \cdots & \frac{df}{da_{m1}} \\ \vdots & \ddots & \vdots \\ \frac{df}{da_{1n}} & \cdots & \frac{df}{da_{mn}} \end{bmatrix} \in \mathbb{R}^{n \times m} \tag{A.18}$$

Thus, the matrix products $\frac{df}{dA}A$ and $\frac{df}{da}a$ are dimensionally compatible. Derivatives of matrices are defined by applying the definitions above to the matrix, entry by entry, i.e., for matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$

$$\frac{dA}{dB} = \begin{bmatrix} \frac{da_{11}}{dB} & \cdots & \frac{da_{1n}}{dB} \\ \vdots & \ddots & \vdots \\ \frac{da_{m1}}{dB} & \cdots & \frac{da_{mn}}{dB} \end{bmatrix} \in \mathbb{R}^{mq \times np} \tag{A.19}$$

Bibliography

[1] Gillespie, C. S. *IET Systems Biology* **2009**, *in press.*

[2] Schlick, T. *IMA Volumes in Mathematics and its Applications* **1997**, *82*, 219–247.

[3] Gillespie, D. T. *The Journal of Physical Chemistry* **1977**, *81*(25), 2340–2361.

[4] Gibson, M. A.; Bruck, J. *The Journal of Physical Chemistry A* **2000**, *104*(9), 1876–1889.

[5] Cao, Y.; Gillespie, D. T.; Petzold, L. R. *The Journal of Chemical Physics* **2006**, *124*(4).

[6] Gillespie, F. T. *J Chem Phys* **2001**, *115*, 1716–1733.

[7] Gillespie, D. T. *The Journal of Chemical Physics* **2000**, *113*(1), 297–306.

[8] Gomez-Uribe, C. A.; Verghese, G. C. *The Journal of Chemical Physics* **2007**, *126*(2), 024109.

[9] Goutsias, J. *Biophysical Journal* **2007**, *92*(7), 2350–65.

[10] Petersen, K. B.; Pedersen, M. *The Matrix Cookbook;* Technical University of Denmark, 2008.

[11] Sauro, H. M.; Ingalls, B. *Biophysical Chemistry* **2004**, *109*(1), 1–15.

[12] Gomez-Uribe, C. A.; Verghese, G. C.; Tzafriri, A. R. *The Journal of Chemical Physics* **2008**, *129*(24), 244112.

[13] Perez-Arriaga, I.; Verghese, G.; Schweppe, F. *IEEE Transaction on Power Apparatus and Systems* **1982**, *PAS-101*(9), 3117–3125.

[14] Perez-Arriaga, I.; Verghese, G.; Pagola, F.; Sancha, J.; Schweppe, F. *Automatica* **1990**, *26*(2), 215–231.

[15] Neudecker, H.; Wansbeek, T. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* **1983**, *11*(3), 221–231.

[16] Liu, S. *Linear Algebra and its Applications* **1999**, *289*, 267–277.