Parameter Estimation of Chemical Reaction Networks: The Super-ensemble Approach

by

Yanping Huang

(Under the direction of H. B. Schüttler)

Abstract

This thesis describes a novel Monte Carlo simulation algorithm for the estimation of the model parameters of kinetic rate equation systems, describing biochemical reaction networks; and for the quantitative prediction of the time-dependent behavior of real biochemical systems described by such kinetics models. This simulation method, referred to as the *super-ensemble* approach, combines Monte Carlo sampling of the kinetics model parameter space with a simultaneous Galerkin-type variational Monte Carlo solution of the underlying kinetic rate equation system. Unlike the recently proposed and closely related "standard" ensemble simulation method, the super-ensemble does not rely on the high-volume execution of a conventional serial ordinary differential equation(ODE) solver algorithm, and it is therefore amenable to an efficient scalable parallelization by straightforward time domain decomposition techniques. With minor modifications, the super-ensemble algorithm can also be deployed as a parallelizable variational ODE solution method, in a conventional ODE solver setting where a unique ODE solution is sought for given initial conditions and given rate functions. Test applications of the super-ensemble algorithm in both ODE solver mode and in parameter estimation mode, for a simple enzyme catalysis model, will be discussed.

Index words: Monte Carlo, Biochemical Network, Ensemble Method

Parameter Estimation of Chemical Reaction Networks: The Super-ensemble Approach

by

Yanping Huang

B.S., Zhejiang University, P.R.China, 2005

A Thesis Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Master of Science

Athens, Georgia

2007

Parameter Estimation of Chemical Reaction Networks:
The Super-ensemble Approach

by

Yanping Huang

Approved:

Major Professor:	H. B. Schüttler
Committee:	Jonathan Arnold William M. Dennis David P. Landau

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2007

Acknowledgments

My chief acknowledgment is to my advisor, Dr. H. B. Schüttler, for his invaluable guidance, encouragement and insight during this research work. He not only taught me how to conduct research in this multidisciplinary area but also how to enjoy it. I have learned how to pay attention to details, use clear, precise, consistent notation and terminology, analyze problems thoroughly, and communicate efficiently. I especially would like to express my appreciation for his support and advising for my career development.

I also would like to thank Dr. Jonathan Arnold, Dr. William M. Dennis and Dr. David P. Landau for serving on my advisory committee. I greatly appreciate the Department of Physics and Astronomy not only for granting a teaching assistantship, but also the great opportunity to take courses with so many great professors. I also wish to thank Dr. S. Tsai, Mike Caplinger and Jeff Deroshia for being always so patient and willing to help me with all kinds of problems with computing facilities. I would like to thank Dr. David Lowenthal, the instructor for the course of *Parallel and Distributed Computing* of Computer Science for his stimulating discussion during this research work. I also greatly appreciate the inspiring discussion with Dr. Wenduo Zhou, Xiaojia Tang and my fellow graduate students in the Department of Physics and Astronomy.

I greatly appreciate Dr. H. B. Schüttler, for granting a research assistantship. This work was supported by in part by the U.S. National Science Foundation under NSF Grant No. BES-0425762. Computing resources from the University of Georgia Research Computing Center are also acknowledged.

Finally, but most importantly, I would like to thank my family. They even don't understand one single word in this dissertation, but without them, this dissertation never would have been written. They give me a life and teach me how to live it.

Table of Contents

Chapter 1

Introduction

Chemical reaction networks provide us a simple and general framework for quantitatively modeling a biological system's dynamic behavior in terms of the time-dependent concentrations of molecular species. These time-dependent concentrations of molecular species in a reaction network can be expressed as a system of coupled ordinary differential equations (ODE), the kinetic rate equations. In principle, such an ODE model of a biochemical network may enable us to describe how biochemistry and genetic activity of a cell evolves as a function of time. It has been successfully applied to systems like metabolic networks, signal transduction, and cell cycle models [13, 19]. In a deterministic modeling approach, given all rates coefficients and initial concentrations of the network, the chemical network's behavior will be completely determined by a unique solution of its ODE system.

To estimate the relevant kinetic rate equation model parameters is a crucial step in quantitatively describing and predicting both the dynamics and long-term steady state behavior of a biochemical system, which, in turn, can promote further understanding of biological mechanisms at the systems level. However, experimental biologists working with real biological networks face the problem that the model parameters are mostly unknown, and the experimental data are typically noisy and not available in adequate amounts for quantitative studies. This makes the identification of model parameters a formidable problem. The ensemble method, using a combination of Monte Carlo simulation and ODE solving techniques, has been proposed to sidestep this problem [2, 4, 20]. The essential idea is not to try to identify one unique set of model parameters, but rather generate a statistical ensemble of candidate models constrained by the available experimental data, in such a manner that those models yield model predictions most consistent with the experimental data. This is achieved by sampling an ensemble probability distribution function that measures the goodness of fit of the model predictions with respect to the given experimental data set, using Monte Carlo methods. Given such a sample of model parameter sets, this approach can then predict ensemble averages along with their higher moments and statistical distribution. Move importantly, it enables us to predict outcomes, and guide the design of, new experiments which can then further reduce the uncertainties of model prediction [6].

In this thesis, I will develop and explore a new computational Monte Carlo-based approach towards kinetic rate equation ensemble modeling of biological networks referred to as the *super-ensemble* method. In contrast to the existing “standard” ensemble approach [2, 4, 20], the super-ensemble does not require a completely *de novo* numerical re-solution of the underlying ODE system every time an ensemble Monte Carlo update is being executed. Instead, the super-ensemble approach combines the ensemble Monte Carlo sampling of the kinetics model parameter space with a variational Monte Carlo exploration of the ODE solution space based on a Galerkin-type cost functional [8].

The standard ensemble method [2, 4, 20] employs conventional numerical ODE solvers, such as the Runge-Kutta and the backward differentiation methods, which have to be executed on the order of 10^6 to 10^8 times during a typical simulation. Since these ODE solving algorithms are intrinsically *serial* in nature, the standard ensemble method does not allow for the implementation of efficient parallelization strategies. By contrast, the super-ensemble Monte Carlo algorithm immediately lends itself to an obviously scalable time domain decomposition parallelization approach.

With a very simple modification in its Monte Carlo initialization and an approximate choice of a virtual experimental input data set, the super-ensemble algorithm can also be employed as a parallelizable ODE solution method in a conventional ODE solver setting. In this setting all ODE initial conditions and all rate function parameters are given, and the super-ensemble algorithm then produces a reasonable variational approximation to the unique exact ODE solution.

The sections of this thesis are organized as follows: in Chapter 2, I introduce the general kinetics rate equation formalism for the modeling of chemical and biological reaction networks and a simple enzyme catalysis kinetics model to illustrate this formalism. In Chapter 3, I briefly recapitulate the standard ensemble method, and I then describe the proposed new super-ensemble formalism. In Chapter 4, I reformulate the kinetics ODE system so that the super-ensemble becomes fully parallelizable, with multiple processors being able to performing multiple Monte Carlo updates simultaneously. In Chapter 5, I describe the generation of virtual experimental data sets and the detailed Monte Carlo protocols which will be used to test the performance of the super-ensemble for the simple biochemical enzyme catalysis kinetics model introduced in Chapter 2. In Chapter 6, I present the test results which I have obtained, with my super-ensemble Monte Carlo code for a simple enzyme catalysis model. Application of the code in both variational ODE solver mode and in full ensemble simulation mode will be discussed. In Chapter 7, I present a brief summary and concluding remarks.

Chapter 2

Kinetics ODE Models of Chemical and Biological Network

2.1 A simple Example: The Enzyme Model

Biological systems can be viewed as chemical reaction networks[3]. Here I describe a simple reaction network for a typical enzymatic reaction in a biological system to exemplify the reaction kinetics modeling of such networks. This model consists of four species, the enzyme E, the substrate S, the product P, and the enzyme-substrate complex ES_2, which participate in the following four reactions

$$r = 1 : \quad E + 2S \rightarrow ES_2 \tag{2.1}$$

$$r = 2 : \quad ES_2 \rightarrow E + P \tag{2.2}$$

$$r = 3 : \quad ES_2 \rightarrow E + 2S \tag{2.3}$$

$$r = 4 : \quad P + E \rightarrow ES_2 \tag{2.4}$$

where $r = 1, \ldots, \mathcal{R} = 4$ labels the reactions. This network can be viewed as a simple model for the catalytic conversion of two copies of substrate molecules S into one copy of product molecule P, mediated by enzyme E acting as the catalyst. Note that $r = 3$ is the *backward* reaction to $r = 1$, and $r = 4$ is the *backward* reaction to $r = 2$.

Such a reaction network can also be graphically represented by a reaction network diagram, as shown and explained in Figure 2.1 [20]. These types of network graphs, consisting of two type of vertices (boxes and circles) with directed edges (arrows) connecting only pairs of different types of vertices, are also referred to as Petri nets in the graph theory

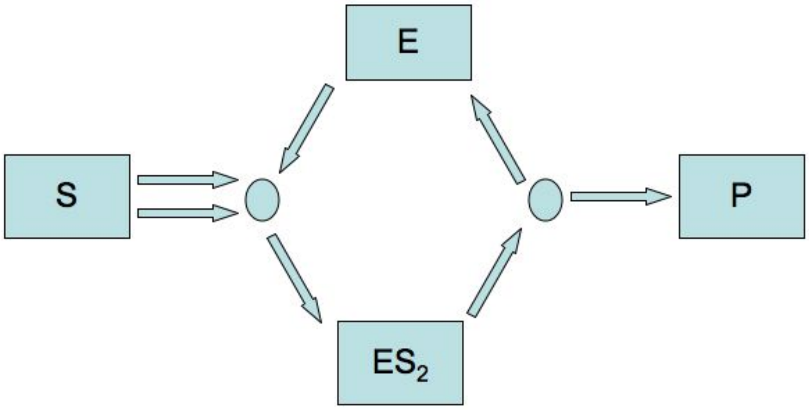


Figure 2.1: Graphical Representation of the reaction network defined in Equation 2.1 - 2.4. Each rectangular box represents a molecular species. Each circle represents two possible reactions: a forward reaction, proceeding in the direction of the arrow, and a backward reaction, proceeding against the arrow direction. The number of arrows drawn from a species box or from a reaction circle indicates the number of molecules of that species entering or leaving the forward reaction, respectively. For the corresponding backward reaction, molecules are entering and leaving against the respective arrow direction.

literature [16]. A number of qualitative graph theoretical analysis tools, based solely on the network topology, have been developed. The aim of the kinetics ensemble modeling approach is to go beyond purely topological considerations in order to elucidate the kinetic-based time evolution of these systems in more quantitative detail.

The kinetics of each reaction component in such a network can be characterized by the so-called kinetic rate coefficients. For this enzyme model, the corresponding four reaction rate coefficients are denoted by $\theta_1, \ldots, \theta_4$. These rate coefficients govern the kinetics of this system, described by coupled set of ODEs. Assuming mass balance kinetics [17, 18, 12], the coupled ODEs for this enzyme model are:

$$\frac{d[E]}{dt} = -\theta_1[E][S]^2 + \theta_2[ES_2] + \theta_3[ES_2] - \theta_4[P][E] \tag{2.5}$$

$$\frac{d[ES_2]}{dt} = \theta_1[E][S]^2 - \theta_2[ES_2] - \theta_3[ES_2] + \theta_4[P][E] \tag{2.6}$$

$$\frac{d[S]}{dt} = -2\theta_1[E][S]^2 + 2\theta_3[ES_2] \tag{2.7}$$

$$\frac{d[P]}{dt} = \theta_2[ES_2] - \theta_4[P][E] \tag{2.8}$$

In the simulations described later, I have simplified the enzyme model somewhat by assuming that the reaction $ES_2 \to E + P$ is irreversible and therefore set

$$\theta_4 = 0 \tag{2.9}$$

2.2 General Kinetics ODE Formalism for Reaction Networks

In general, the kinetic ODEs in a reaction network have the form

$$\dot{\psi}_s(t) = f_s(\boldsymbol{\psi}\ (t), t) \tag{2.10}$$

where

$\dot{\psi}_s(t) := d\psi_s/dt$, the first time derivative of the concentration of species s.

$s = 1, \ldots, \mathcal{S}$, is the index of species in the reaction network.

$\mathcal{S}$ is the number of species in a reaction network.

$\boldsymbol{\psi}\ (t) := (\ldots, \psi_s(t), \ldots)$, a $\mathcal{S}$-dimensional vector of all species concentrations.

and the rate functions follow mass balance kinetics

$$f_s(\boldsymbol{\psi}\ , t) = \sum_{r=1}^{\mathcal{R}} g_{sr}\gamma_r(t)\theta_r \prod_{s'=1}^{\mathcal{S}} (\psi_{s'})^{h_{s'r}} \tag{2.11}$$

where

$r = 1\ \ldots\ , \mathcal{R}$, is the index of reactions in the network.

$\mathcal{R}$ is the number of reactions.

$\boldsymbol{\Theta} = (\ldots, \theta_r, \ldots)$ is the $\mathcal{R}$-dimensional vector of the reaction rate coefficients for reaction network.

$h_{s'r}$ is the matrix of stoichiometric input coefficient, i.e., the number of copies of molecules of s' entering reaction r.

g_{sr} is the matrix of stoichiometric net production coefficients of species s in reaction r.

$\gamma_r(t)$ are externally controlled modulation factors, e.g. due to externally controlled time-dependent thermal cycling, light-exposure of feeding schedules.

In this thesis I only consider the situation without external modulation, so $\gamma_r(t) = 1$, and the rate functions f_s do not have any explicit time-dependence, i.e., $f_s = f_s(\psi)$ only. For the enzyme model example, the values of g_{sr} and h_{sr} are shown in Table 2.1. Note that, in the general case (Equation 2.11), if $\bar{r}$ is the backward reaction of r, then

$$g_{sr} = h_{s\bar{r}} - h_{sr} \tag{2.12}$$

Table 2.1: Table of coefficients for Enzyme model

(a) Table of h_{sr} for Enzyme model, where h_{sr} is the stoichiometric input coefficient

Species s \ Reaction r	1	2	3	4
E	1	0	0	1
ES_2	0	1	1	0
S	2	0	0	0
P	0	0	0	1

(b) Table of g_{sr} for Enzyme model, where g_{sr} is the net stoichiometric production coefficient

Species s \ Reaction r	1	2	3	4
E	-1	1	1	-1
ES_2	1	-1	-1	1
S	-2	0	2	0
P	0	1	0	-1

and hence,

$$g_{s\bar{r}} = -g_{sr} \tag{2.13}$$

Given all rate coefficients θ_r, all modulation factors $\gamma_r(t)$ and all initial concentrations $\psi_s(t_0)$ at some initial time t_0, the solution to the ODE system is uniquely determined and can be obtained by conventional numerical ODE solving methods [5, 14].

Chapter 3

The super-Ensemble Method

3.1 The Standard Ensemble Monte Carlo(MC) Method

As in the study of systems biology, most realistic biological networks, for the foreseeable future, are likely to be parameter rich and data poor. This makes identification of unique model parameters extremely difficult. The standard ensemble approach [2, 4, 20] was developed to circumvent this problem based on ideas that are borrowed from statistical mechanics. Instead of trying to identify one unique parameterization model, it aims to identify an ensemble of models consistent with the available experimental data and it uses MC simulation techniques to generate random samples of model parameterizations that represent this ensemble.

The standard ensemble method starts from a probability distribution function on the space of all model parameters $\mathbf{\Gamma}$, given by

$$Q(\mathbf{\Gamma}) = \frac{1}{\mathcal{Z}}\exp[-H_{\mathrm{Q}}(\mathbf{\Gamma})] \tag{3.1}$$

here $\mathcal{Z}$ is the normalization factor $\mathcal{Z} = \int_{\mathbf{\Gamma}} \exp[-H_{\mathrm{Q}}(\mathbf{\Gamma})]$, $\mathbf{\Gamma}$ is the vector of all unknown model parameters, and $\int_{\mathbf{\Gamma}}$ denotes integration over $\mathbf{\Gamma}$ space. The $\mathbf{\Gamma}$ consists of all the rate coefficients θ_r and initial concentrations

$$\mathbf{X} := (\ldots, X_s, \ldots) := (\ldots, \psi_s(t_0), \ldots) \tag{3.2}$$

which are required to specify a unique solution to the ODE model, i.e.,

$$\mathbf{\Gamma} := (\Gamma_1, \ldots, \Gamma_{\mathcal{M}}) = (X_1, \ldots X_{\mathcal{S}}, \theta_1, \ldots, \theta_{\mathcal{R}}) \tag{3.3}$$

where

$$
\begin{aligned}
\mathcal{M} &= \mathcal{S} + \mathcal{R} && (3.4)\\
\Gamma_m &= \begin{cases} X_m & 1 \le m \le \mathcal{S} \\ \theta_{m-\mathcal{S}} & \mathcal{S}+1 \le m \le \mathcal{R} \end{cases} && (3.5)
\end{aligned}
$$

More generally, the model parameter vector $\mathbf{\Gamma}$ could also include unknown unit conversion factors or unknown stoichiometric coefficients h_{sr}, in cases where the reaction network topology is partially unknown. In this thesis I will not consider such cases, i.e., $\mathbf{\Gamma}$ will still be given by Equation 3.5.

The so-called fictitious energy function $H_Q(\mathbf{\Gamma})$ is given by

$$H_{\mathrm{Q}}(\mathbf{\Gamma}) = \sum_{n,s} (\psi_s(\bar{t}_n; \mathbf{\Gamma}) - Z_{n,s})^2 / (2\sigma^2_{n,s}) \tag{3.6}$$

where

n is the index of the experimental observation time points, $n = 1 \ldots \mathcal{N}$.

$\mathcal{N}$ is the number of experimental observation time points.

$\bar{t}_n$ are the experimental observation time points.

$Z_{n,s}$ is the experimentally observed concentration of species s at time $\bar{t}_n$.

$\sigma_{n,s}$ is the standard deviation of the experimental data point $Z_{n,s}$ of species s at time $\bar{t}_n$.

Note that $\psi_s(t; \mathbf{\Gamma})$ for species s at time t denotes the solution of ODE model, obtained with the model parameter vector $\mathbf{\Gamma}$. I now explicitly include the dependence of the ODE solution ψ_s on the model parameters in the argument of $\psi_s(t; \mathbf{\Gamma})$. Note that $H_Q(\mathbf{\Gamma})$ is essentially just the standard χ^2-function used in least-square fitting procedures. $H_Q(\mathbf{\Gamma})$ is a measure of how well the model prediction $\Psi_s(\bar{t}_n, \mathbf{\Gamma})$ for a given choice of model parameters $\mathbf{\Gamma}$ match as the experimental data $Z_{n,s}$. If a sufficient amount of data $Z_{n,s}$ of sufficient quality, i.e. “small”

$\sigma_{n,s}$, is available, one can simply perform a non-linear least-squares fit to extract values for the model parameters $\mathbf{\Gamma}$ from the data, by minimizing $H_Q(\mathbf{\Gamma}) := \frac{1}{2}\chi^2(\mathbf{\Gamma})$ with respect to $\mathbf{\Gamma}$.

However, in typical biological applications, the data are noisy and for many species s only a few or no data points at all have been measured. As a consequence, the (absolute) minimum of $H_Q(\mathbf{\Gamma})$ is either poorly defined, or worse, there is no unique local minimum that clearly defines "the" best fit to the data; rather, there is an entire manifold (hypersurface) of points in $\mathbf{\Gamma}$-space such that all $\mathbf{\Gamma}$-points on that manifold give an equally good fit to the data.

The basic idea of the ensemble approach is therefore to generate not one unique $\mathbf{\Gamma}$ providing "the" best fit to the data, but rather a statistical sample of all $\mathbf{\Gamma}$ that are consistent with the data. The probability distribution function $Q(\mathbf{\Gamma})$ in Equation 3.1 defines that $\mathbf{\Gamma}$-sample. If a $\mathbf{\Gamma}$ gives a good fit to the data, $H_Q(\mathbf{\Gamma})$ is "small"; therefore $\mathbf{\Gamma}$ has a high probability $Q(\mathbf{\Gamma})$ for being included in the sample; $\mathbf{\Gamma}$ giving a bad fit to the data have a "large" $H_Q(\mathbf{\Gamma})$ and therefore a low probability. The standard ensemble method then proceeds by calculating the ensemble average over $\mathbf{\Gamma}$ space, i.e., for some quantities $A(\mathbf{\Gamma})$, one wants to calculate

$$< A(.) >_{\mathrm{Q}} := \int_{\mathbf{\Gamma}} A(\mathbf{\Gamma}) Q(\mathbf{\Gamma}) \tag{3.7}$$

where $\int_{\mathbf{\Gamma}}$ again stands for integration over $\mathbf{\Gamma}$. Such an average can then be used to predict the outcome of future experiments. Also, the ensemble standard deviation(ESD) of A

$$\sigma[A] := (< [A(.) - < A(.) >_{\mathrm{Q}}]^2 >_{\mathrm{Q}})^{1/2} \tag{3.8}$$

can be used to quantify the uncertainty of such predictions.

The evaluation of such ensemble averages $< \ldots >_Q$ is carried out by standard Markov Chain Monte Carlo(MC) approaches [10], such as the Metropolis method. In such an MC approach one numerically performs a random walk in $\mathbf{\Gamma}$-space in such a way that any $\mathbf{\Gamma}$ is visited by the random walk with a probability distribution that equals $Q(\mathbf{\Gamma})$, after a sufficiently large number of random steps. For example in the Metropolis MC method [10], the random walk proceeds by the following random updating scheme:

1. Given $\mathbf{\Gamma} = (\Gamma_1, \ldots, \Gamma_{\mathcal{M}})$, select a random m from $(1, \ldots, \mathcal{M})$

2. Propose a change of Γ_m to

$$\Gamma'_m = \Gamma_m + \Delta_m \times (2u - 1) \tag{3.9}$$

   where u is a random number drawn uniformly from $[0, 1]$ and Δ_m is a stepwidth parameter to be adjusted for optimal performance, as described later (in Section 5.4).

3. Set

$$\mathbf{\Gamma}' = (\Gamma_1, \ldots, \Gamma_{m-1}, \Gamma'_m, \Gamma_{m+1}, \ldots, \Gamma_{\mathcal{M}}) \tag{3.10}$$

   and accept the proposed change $\mathbf{\Gamma} \to \mathbf{\Gamma}'$ with the Metropolis acceptance probability

$$\begin{aligned} p_Q(\mathbf{\Gamma} \to \mathbf{\Gamma}') &= \min(1, \frac{Q(\mathbf{\Gamma}')}{Q(\mathbf{\Gamma})}) && (3.11) \\ &= \min(1, \exp[-\Delta H_Q(\mathbf{\Gamma} \to \mathbf{\Gamma}')]) && (3.12) \end{aligned}$$

   where $\Delta H_Q(\mathbf{\Gamma} \to \mathbf{\Gamma}') = H_Q(\mathbf{\Gamma}') - H_Q(\mathbf{\Gamma})$.

4. If the change $\mathbf{\Gamma} \to \mathbf{\Gamma}'$ is accepted, set $\mathbf{\Gamma}$ to $\mathbf{\Gamma}'$, otherwise leave $\mathbf{\Gamma}$ unchanged.

This four-part updating step is repeated sufficiently many times until the random $\mathbf{\Gamma}$ produced are distributed according to $Q(\mathbf{\Gamma})$ [10].

3.2 The Super-Ensemble Method

Although successful in the reconstruction of smaller biological networks such as the *qa* gene cluster [11, 2] and the biological clock [2], the kinetics model ensemble method in its current standard form [20, 2, 4] has two drawbacks in large-systems applications: it is highly CPU-time intensive and of limited parallelizability. The network's kinetic rate equations have to be re-solved for each proposed ensemble Monte Carlo (MC) updating step, *i.e.*, typically 10^6 -10^8 times per simulation. Conventional ordinary differential equation (ODE) solvers, such as adaptive Runge-Kutta or backward differentiation [14, 5], are serial and slow: the "next"

ODE step at time $t+h$ requires a complete knowledge of the solution at the previous t; and the numerics operates on an “all-or-nothing” principle in that the ODE solver either generates a highly accurate solution of typically $10^{-4}\%$ relative error or better or else succumbs to numerical instability. This level of accuracy is “overkill” in kinetics model ensemble MC applications, where a “quick and dirty” ODE solution with, say, up to 10% relative error would suffice, since the error bars of experimental t-dependent concentration data entering into the ensemble probability distribution $Q(\mathbf{\Theta})$ are typically at or above the 10% level. Also, the equilibration of conventional MC updating schemes, such as Metropolis [10], are intrinsically serial in a kinetics model ensemble setting: the “next” update cannot be done before the last one is completed, and the network's kinetics ODE system constitutes a highly non-local and heterogeneous coupling environment in the network-spatial and temporal domain. This non-locality of the standard ensemble energy function H_{Q} precludes effective domain decomposition parallelization strategies.

The basic idea of the super-ensemble is to combine the Monte Carlo exploration of the model parameter $\mathbf{\Gamma}$ space and the solution of ODE system into a single MC procedure. To do this, I expand the space of MC variables to include both the orginal $\mathbf{\Theta}$-vector of the standard ensemble method and an additional vector of variational variables denoted by $\mathbf{Y}$, which are used to represent an approximate variational solution to the ODE system of the reaction network, denoted by $\mathbf{\Psi}(t;\mathbf{Y}) = (\ldots, \Psi_s(t;\mathbf{Y}), \ldots)$, as defined later.

On this expanded ($\mathbf{\Theta}$,$\mathbf{Y}$)-space, I define an expanded fictitious energy function:

$$H(\mathbf{\Theta},\mathbf{Y}) = \beta_{\mathrm{X}} H_{\mathrm{X}}(\mathbf{Y}) + \beta_{\mathrm{K}} H_{\mathrm{K}}(\mathbf{\Theta},\mathbf{Y}), \tag{3.13}$$

here H is the sum of two weighted contributions H_{X} and H_{K}, with positive weight factors β_{X} and β_{K}, respectively. In the first piece of H, H_{X}, the so-called experimental part of the energy, is the original standard ensemble energy function with the exact kinetic solution $\psi_s(t,\mathbf{\Gamma})$ replaced by the approximate variational solution $\Psi_s(t;\mathbf{Y})$, i.e.,

$$H_{\mathrm{X}}(\mathbf{Y}) \quad = \quad \sum_{n,s} [\mathbf{\Psi}(\bar{t}_n;\mathbf{Y}) - Z_{n,s}]^2/(2\sigma_{n,s}^2) \tag{3.14}$$

$$:= \sum_{n,s} (\rho^{(\mathrm{X})}_{n,s})^2, \tag{3.15}$$

where the residues $\rho^{(\mathrm{X})}_{n,s}$ are given by

$$\rho^{(\mathrm{X})}_{n,s} := (\Psi_s(\bar{t}_n; \mathbf{Y}) - Z_{n,s})/(\sqrt{2}\sigma_{n,s}) \tag{3.16}$$

The second piece of $H(\mathbf{\Theta}, \mathbf{Y})$ is the so-called kinetic part of the energy function, defined as

$$H_{\mathrm{K}}(\mathbf{\Theta}, \mathbf{Y}) = \sum_{s=1}^{\mathcal{S}} \sum_{k=1}^{\mathcal{K}} (\dot{\Psi}_s(\hat{t}_k; \mathbf{Y}) - f_s(\mathbf{\Psi}(\hat{t}_k; \mathbf{Y}), t; \mathbf{\Theta}))^2 \tag{3.17}$$

where $\mathcal{K}$ is the number of check points, $\hat{t}_k$ is in a grid of time check points distributed over the simulation time interval, to be defined later. The rate functions f_s are defined as in Equation 2.11,with the dependence on the $\mathbf{\Theta}$-variables now explicitly shown in the argument list of $f_s(\mathbf{\Psi}, t; \mathbf{\Theta})$.

The vector of variational ODE solutions $\mathbf{\Psi}$ comprises the time-dependent variational solutions $\Psi_s(t; \mathbf{Y})$ for all species s

$$\mathbf{\Psi}(t; \mathbf{Y}) = (..., \Psi_s(t; \mathbf{Y}), ...) \tag{3.18}$$

and

$$\mathbf{Y} = (..., y_{i,s}, ...) \tag{3.19}$$

The Ψ_s are given in terms of variational amplitude variables $y_{i,s}$ by:

$$\Psi_s(t; \mathbf{Y}) = \sum_i y_{i,s} \Phi_i(t), \tag{3.20}$$

where the $\Phi_i(t)$ are an appropriately chosen set of basis functions, as described in more detail below in Section 3.3, and $i = 0, ..., \mathcal{I}$ with $\mathcal{I} + 1$ being the number of basis functions in the set $\{\Phi_i\}$.

The kinetic part of the fictitious energy, H_{K}, is the sum of the squared residues of the kinetic rate equations,

$$H_{\mathrm{K}} = \sum_{s,k} (\rho^{(\mathrm{K})}_{s,k})^2 \tag{3.21}$$

$$\rho^{(\mathrm{K})}_{s,k} = \dot{\Psi}_s(\hat{t}_k, \mathbf{Y}) - f_s(\mathbf{\Psi}(\hat{t}_k; \mathbf{Y}), \hat{t}_k; \mathbf{\Theta}) \tag{3.22}$$

that is the difference between the right hand side and left hand side of Equation 2.10, evaluated at the check point time $\hat{t}_k$. If $\Psi_s(t;\mathbf{Y})$ were to obey exactly the ODE system, then all $\rho_{s,k}^{(\mathrm{K})}$ would be zero, and H_{K} would be minimized with zero value. In the actual super-ensemble calculation, $\Psi_s(t;\mathbf{Y})$ will not exactly obey the ODE. However, by minimizing H_{K} with respect to $\mathbf{Y}$, I expect to generate a reasonably accurate approximate solution $\Psi_s(t;\mathbf{Y})$ to the ODE system. The larger the number of basis functions $\Phi_i(t)$ included in Equation 3.20 to represent $\Psi_s(t;\mathbf{Y})$ and the denser the check point grid $\hat{t}_k$ being used in H_{K}, the more accurately I expect $\Psi_s(t;\mathbf{Y})$ to approximate the exact solution to ODE system (Equation 2.10) after minimizing H_{K}.

This variational approach has been frequently used for the numerical solution of partial differential and integral equations in higher dimensions and is known as the Galerkin approach in the literature [8]. However, this approach does not appear to have been used for ordinary differential equations. This is probably due to the fact that highly accurate sequential ODE solving algorithms are available, which are very efficient when only a few, high accuracy solutions of an ODE system are required, for given initial conditions. In the present context, as noted in the earlier part of this section, I only need low-accuracy approximate ODE solutions, but I need *many* of them, and I need a parallelizable method for calculating them efficiently. Also, I am *not* given the initial conditions of the ODE system, but rather the ODE solutions are constrained by the requirement that the solution ψ optimally match the experimental data $Z_{n,s}$ which are spread out along the entire simulation time interval $[t_0, t_{\mathcal{I}}]$.

Furthermore, I would like to be able to take advantage of the fact that the *next* ODE solution, during the MC random walk in the model parameter space, is usually similar to the *previous* ODE solution already visited in the solution space. Therefore it is rather wasteful to re-calculate the next ODE solution “from scratch” during each MC update, as is done in the standard ensemble approach. By contrast, in the variational approach I am fully exploiting the proximity of the next and previous ODE solution during the Monte Carlo random walk.

Note here that minimizing $H_{\mathrm{K}}(\mathbf{\Theta}, \mathbf{Y})$ with respect to $\mathbf{Y}$ will generally not produce a unique approximate ODE solution, but rather an entire continuum of ODE solutions, corresponding to different choices of the initial condition $\mathbf{X}$, in Equation 3.2. A unique ODE solution would be selected from this solution continuum by performing an H_{K} minimization subject to the constraint

$$\mathbf{\Psi}(t_0; \mathbf{Y}) = \mathbf{X} \tag{3.23}$$

However, for the task of extracting information about the ODE model parameters from experimental data I instead constrain the ODE solution continuum by adding H_{X} in the fictitious energy $H(\mathbf{\Theta}, \mathbf{Y})$. By minimizing $H(\mathbf{\Theta}, \mathbf{Y})$ with respect to $\mathbf{\Theta}$ and $\mathbf{Y}$, I will then obtain an approximate ODE solution (or a sub-continuum of such approximate ODE solutions) which gives the best possible fit(s) to the experimental data in the limit as

$$\frac{\beta_{\mathrm{K}}}{\beta_{\mathrm{X}}} \to \infty \tag{3.24}$$

In the actual ensemble simulations, I am only interested in finding approximate solutions to the ODE system, with an accuracy not significantly better than the experimental error bars $\sigma_{n,s}$ entering in H_{X}. I can therefore use large, but finite value of $\frac{\beta_{\mathrm{K}}}{\beta_{\mathrm{X}}}$ in the simulations.

Furthermore, the ultimate goal in the ensemble simulations is not to find *the* $(\mathbf{\Theta}, \mathbf{Y})$ giving *the* best possible fit of the model to the experimental data. Rather, I want to generate a random sample of all model vectors $(\mathbf{\Theta}, \mathbf{Y})$, which are consistent with the data. Therefore, I again define an ensemble probability distribution analogous to the standard ensemble distribution, but now on the expanded $(\mathbf{\Theta}, \mathbf{Y})$-space, by

$$\bar{Q}(\mathbf{\Theta}, \mathbf{Y}) = \frac{1}{\bar{\mathcal{Z}}} \exp[-(\beta_{\mathrm{K}} H_{\mathrm{K}}(\mathbf{\Theta}, \mathbf{Y}) + \beta_{\mathrm{X}} H_{\mathrm{X}}(\mathbf{Y}))] \tag{3.25}$$

where

$$\bar{\mathcal{Z}} = \int_{\mathbf{\Theta}} \int_{\mathbf{Y}} \exp[-(\beta_{\mathrm{K}} H_{\mathrm{K}}(\mathbf{\Theta}, \mathbf{Y}) + \beta_{\mathrm{X}} H_{\mathrm{X}}(\mathbf{Y}))] \tag{3.26}$$

If I take

$$\beta_{\mathrm{K}} \to \infty \tag{3.27}$$

and set

$$\beta_{\mathrm{X}} = 1, \tag{3.28}$$

the probability distribution $\bar{Q}(\boldsymbol{\Theta}, \mathbf{Y})$ approximates the original standard ensemble $Q(\boldsymbol{\Theta}, \mathbf{X})$, to within controllable errors due to the finite basis representation. Formally this finite-basis approximation to Q is recovered from $\bar{Q}$ by integrating out the $\mathbf{Y}$-variables, subject to a δ-function constraint factor to enforce the initial conditions $\mathbf{X}$ on the variational functions $\Psi_s(t; \mathbf{Y})$, i.e.,

$$Q(\boldsymbol{\Theta}, \mathbf{X}) \simeq [\int_{\mathbf{Y}} \delta(\mathbf{X} - \boldsymbol{\Psi}(t_0; \mathbf{Y}))\bar{Q}(\boldsymbol{\Theta}, \mathbf{Y})]_{\beta_{\mathrm{K}} \to \infty, \beta_{\mathrm{X}} = 1} \tag{3.29}$$

So, the super-ensemble represents an approximate reformulation of the original standard ensemble.

I again perform a Metropolis random walk over the $(\boldsymbol{\Theta}, \mathbf{Y})$ space to generate a $(\boldsymbol{\Theta}, \mathbf{Y})$ MC sample drawn from $\bar{Q}(\boldsymbol{\Theta}, \mathbf{Y})$. The details of this Metropolis MC updating procedure are entirely analogous to the MC updating in the standard ensemble method, as described above in Section 3.1. The only difference is that now the $(\boldsymbol{\Theta}, \mathbf{Y})$-vector is subject to the MC updates, instead of the $(\boldsymbol{\Theta}, \mathbf{X})$-vector used in the standard approach.

3.3 Finite Element Method and Time Grids

To define the finite element (FE) basis functions [15], I first need to lay out a grid of interpolation time points, defined by

$$t_i = t_0 + i \times h \quad \text{for } i = 0, \ldots, \mathcal{I} \tag{3.30}$$

Although FE bases can be defined for non-equidistant grids, I will use only equidistant grids with a grid spacing

$$h = (t_{\mathcal{I}} - t_0)/\mathcal{I} \tag{3.31}$$

where t_0 and $t_{\mathcal{I}}$ are the initial and final time points of the simulation interval that comprises all experimental observation times $\bar{t}_n$, described in Section 3.1.

I will first consider two different choices of Lagrange FE basis functions, the first order Lagrange FE basis, as shown in Figure 3.1(a):

$$\Phi_i(t) = (1 - |\xi|)\Theta(1 - |\xi|) \tag{3.32}$$

and the second order Lagrange FE basis, as shown in Figure 3.1(b):

$$\Phi_i(t) = \begin{cases} \frac{1}{2}(|\xi| - 1)(|\xi| - 2)\Theta(2 - |\xi|) & \text{for even i} \\ (1 - \xi^2)\Theta(1 - |\xi|) & \text{for odd i} \end{cases} \tag{3.33}$$

where ξ is a re-scaled local time variable

$$\xi = (t - t_i)/h \tag{3.34}$$

and $\Theta(\xi)$ is the Heaviside step function

$$\Theta(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases} \tag{3.35}$$

The variational model solution $\mathbf{\Psi}(t)$ is then represented in terms of $\Phi_i(t)$ by

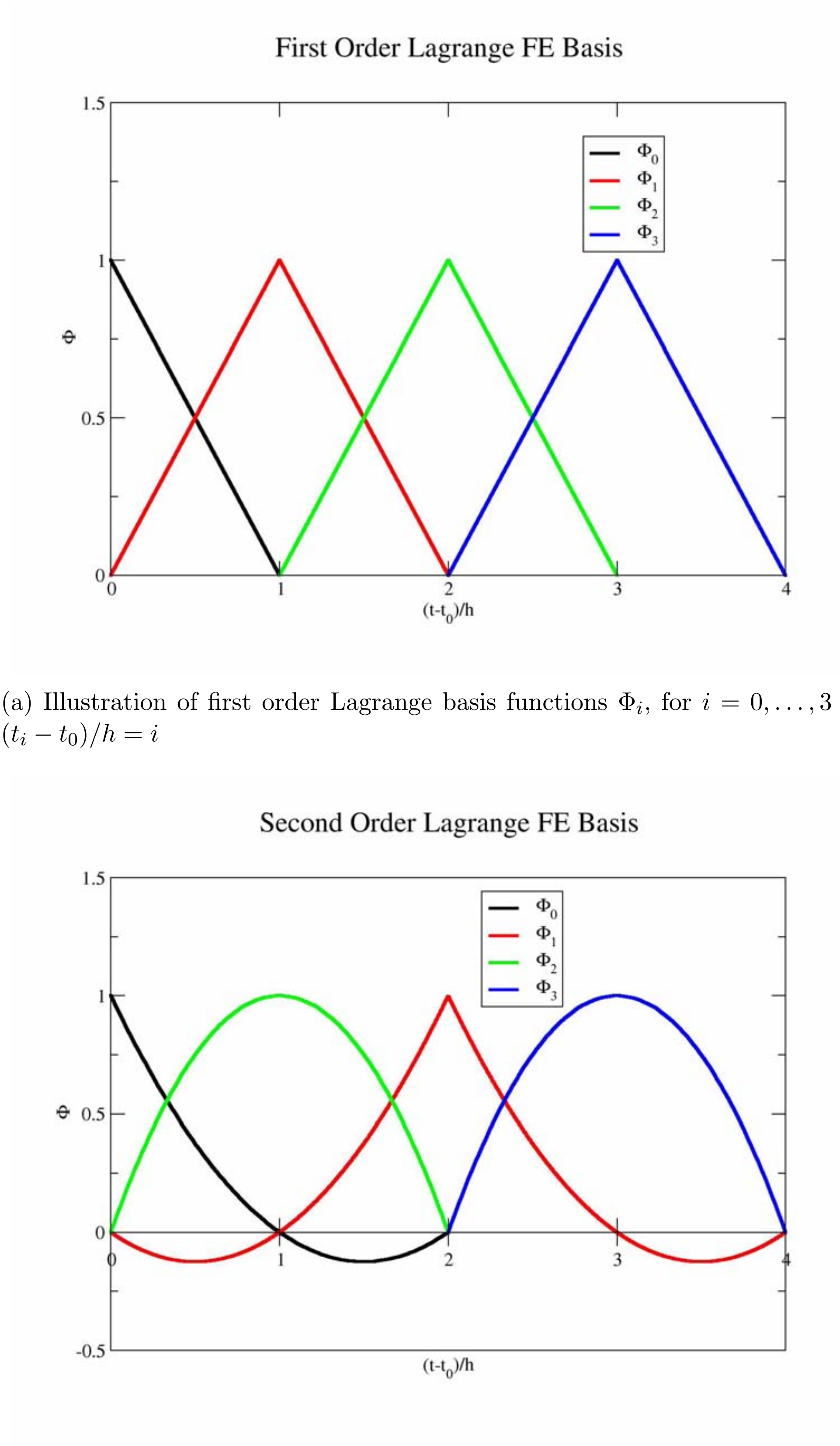
$$\Psi_s(t) = \sum_i y_{i,s}\Phi_i(t) \tag{3.36}$$

so that the $y_{i,s}$ are given by the function values at the corresponding interpolation grid points, i.e..

$$y_{i,s} = \Psi_s(t_i) \tag{3.37}$$

For Hermite FE bases [15], the basis functions acquire an additional index ν with values $\nu = 0, 1$, representing the value and the first order derivative of the interpolated function, i.e, $\Phi_i(t)$ becomes $\Phi_{i,\nu}(t)$ with corresponding local interpolation polynomials $\phi_\nu(\xi)$ so that

$$\begin{aligned} \Phi_{i,\nu}(t) &= \phi_\nu(|\xi|) \times \mathrm{sgn}(\xi)^\nu \times \Theta(1 - |\xi|) && (3.38) \\ \mathrm{sgn}(\xi) &= \begin{cases} 1, & \xi \geq 0 \\ -1, & \xi < 0 \end{cases} && (3.39) \end{aligned}$$



(a) Illustration of first order Lagrange basis functions Φ_i, for $i = 0, \ldots, 3$ with $(t_i - t_0)/h = i$

(b) Illustration of first order Lagrange basis functions Φ_i, for $i = 0, \ldots, 3$ with $(t_i - t_0)/h = i$

Figure 3.1:

for third order Hermite FE basis with polynomials $\phi_\nu(\xi)$.

$$\phi_0(\xi) \quad = \quad 2\xi^3 - 3\xi^2 + 1 \tag{3.40}$$

$$\phi_1(\xi) \quad = \quad 2\xi^3 - 4\xi^2 + 2\xi \tag{3.41}$$

The interpolated function $\Psi_s(t) = \Psi_s^{(0)}(t)$ and its first time derivative $\Psi_s^{(1)}(t)$ are:

$$\Psi_s^{(\mu)}(t) = \frac{d^\mu}{dt^\mu}\Psi_s(t) = \sum_{i,\nu} y_{i,s,\nu}\Phi_{i,\nu}^{(\mu)}(t) \quad \text{for } \mu = 0, 1 \tag{3.42}$$

so that the $y_{i,s,\nu}$ are given by function value and its first order derivative at the interpolation grid points, i.e,.

$$y_{i,s,\nu} = \Psi_s^{(\nu)}(t_i) \quad \text{for } \nu = 0, 1 \tag{3.43}$$

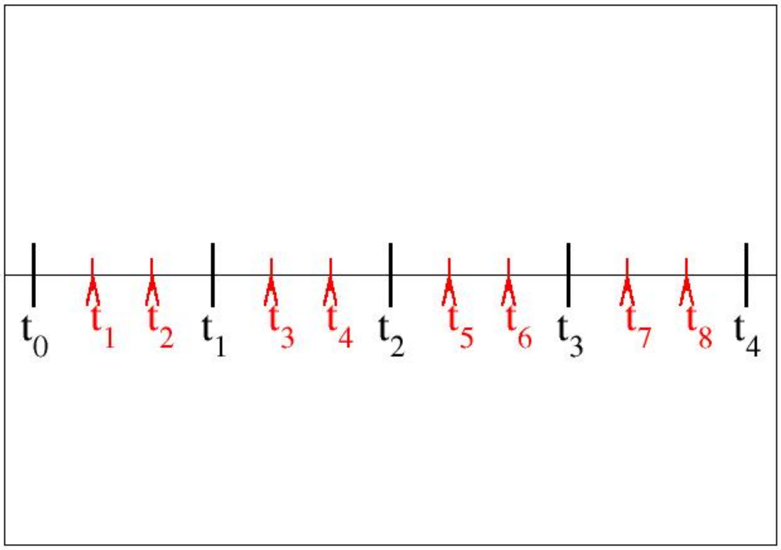


Figure 3.2: Illustration for two different types of grid points: the interpolation grid t_i in black with $i = 0, \ldots, \mathcal{I} = 4$, and the ODE time check point grid $\hat{t}_k$ in red with $k = 1, \ldots, \mathcal{K}$. Note that in this figure, $K_I = 2$, thus I have $\mathcal{K} = K_I \times \mathcal{I} = 8$

Given the interpolation grid t_i I can now also define the time check point grid $\hat{t}_k$ referred to earlier in Section 3.2 in the definition of H_{K}, Equation 3.17. I have developed FE code for all three of the above-described FE basis functions. Preliminary results suggest that the higher order FE bases, i.e., the second order Lagrange and third order Hermite do not substantially improve the quality of the ODE solutions, while at the same time requiring greater computational effort. I will therefore be using only the first order Lagrange FE functions for the test simulations on the enzyme model, discussed below. Since Lagrange FE basis functions have discontinuities in their first derivative at the grid points t_i, I want to construct the $\hat{t}_k$ so as to *not* include any t_i on the interpolation grid. I thus choose a certain number K_I of such $\hat{t}_k$ to be equidistantly spaced in each interpolation interval $[t_{i-1}, t_i]$, but exclude the interval boundaries t_{i-1} and t_i, as shown in Figure 3.2, for $K_I = 2$ and $\mathcal{I} = 4$. Formally, the check point grid $\hat{t}_k$ is then defined by

$$\hat{t}_1 = \frac{h}{K_I + 1} = \hat{h} \tag{3.44}$$

$$\hat{t}_{k+1} = \begin{cases} \hat{t}_k + \hat{h} & \text{if } \hat{t}_k + \hat{h} \neq t_i \text{ for all } i \\ \hat{t}_k + 2\hat{h} & \text{if } \hat{t}_k + \hat{h} = t_i \text{ for } i \neq \mathcal{I} \end{cases} \tag{3.45}$$

In all test simulations discussed in Chapter 6, I have used $K_I = 1$ check point per interpolation interval.

Chapter 4

Time-Domain Parallelization

4.1 Fully Parallelizable Reformulation of the super-ensemble Method

The super-ensemble method outlined in the previous section is well suited for implementation on a serial computer. Unfortunately, it still does not lend itself to an efficient, scalable parallelization. This is due to the fact that the energy function H depends non-locally on all reaction rates variables $\mathbf{\Theta}$. That is to say, a Metropolis update of a single θ_r variable changes the values of the residues $\rho^{(\mathrm{K})}_{s,k}$ at *all* time check points $\hat{t}_k$. This temporal non-locality prevents me from implementing an efficient domain decomposition along the time axis.

One could of course try to do a domain decomposition in the species (s) domain. However, this type of decomposition would present serious difficulties with load balancing, since the reaction networks I consider are typically very in-homogeneous, i.e., there is no regularity in the network topology like the regular crystal lattices one encounters in solid state physics.

To make the super-ensemble method parallelizable, I remove this time non-locality from the energy function by introducing additional fictitious molecular species into the network labeled by species index values

$$s = \mathcal{S} + 1, \ldots, \mathcal{S} + \mathcal{R} \tag{4.1}$$

with corresponding time dependent concentration functions $\psi_s(t)$. Each of these fictitious $\psi_s(t)$ is associated with a corresponding reaction

$$r = s - \mathcal{S} \tag{4.2}$$

or

$$s = \mathcal{S} + r \tag{4.3}$$

and they are incorporated into the ODE model by modifying the ODE system

$$\dot{\psi}_s(t) = f_s(\psi(t), t) \tag{4.4}$$

for $s = 1, \ldots, \mathcal{R} + \mathcal{S}$ with new rate functions f_s as follows:

$$f_s(\boldsymbol{\psi}\ , t) = \begin{cases} \displaystyle\sum_{r=1}^{\mathcal{R}} g_{sr} \gamma_r(t) \psi_{\mathcal{S}+r}(t) \prod_{s'=1}^{\mathcal{S}} \psi_{s'}(t)^{h_{s'r}} & \text{for } s = 1, \ldots, \mathcal{S} \\ 0 & \text{for } s = \mathcal{S}+1, \ldots, \mathcal{S}+\mathcal{R} \end{cases} \tag{4.5}$$

Because of Equation 4.5, the fictitious $\psi_s(t)$ for $s = \mathcal{S} + 1, \ldots, \mathcal{S} + \mathcal{R}$ have no time dependence, i.e.,

$$\psi_s(t) = \psi_s(t_0) = X_s = const \tag{4.6}$$

for all $t \in [t_0, t_{\mathcal{I}}]$.

Also note that in Equation 4.5, the fictitious $\psi_{\mathcal{S}+r}$ has simply replaced the rate coefficient θ_r in the original rate function f_s in Equation 2.11. Therefore, if I simply set the initial value of $\psi_{\mathcal{S}+r}(t)$ to the rate coefficient value θ_r,

$$\psi_{\mathcal{S}+r}(t_0) = \theta_r \tag{4.7}$$

then the resulting ODE system Equation 4.4 will have exactly the same solution, as the original ODE system in Equation 2.10. The modified ODE model Equation 4.4, 4.5 and 4.7 is mathematically equivalent to the original Equation 2.10 and 2.11.

However, I can now use this expanded fictitious species reformulation of the ODE model, Equation 4.4 and 4.5 as the basis for a super-ensemble energy function H with a probability distribution $\bar{Q}$ in complete analogy to the construction of the super-ensemble method in Section 3.2. Using the expanded ODE model with fictitious species $\psi_{\mathcal{S}+r}(t)$ has the great advantage that the rate coefficients θ_r and their non-local effects on the energy function H have been completely eliminated from the ensemble distribution: the θ_r variables of the original super-ensemble method have been replaced by the amplitude variables $y_{i,\mathcal{S}+r}$ of the variational fictitious species concentration $\Psi_{\mathcal{S}+r}(t;\mathbf{Y})$. In the limit $\beta_{\mathrm{K}}/\beta_{\mathrm{X}} \to \infty$, or practically speaking, sufficiently large finite $\beta_{\mathrm{K}}/\beta_{\mathrm{X}}$, the ODE solution constraint on $\Psi_{\mathcal{S}+r}(t;\mathbf{Y})$ will be enforced by the ensemble so that the $\Psi_{\mathcal{S}+r}(t;\mathbf{Y})$ sampled in the simulation are approximately constant and their constant values represent the original rate coefficient θ_r.

The expanded ODE model $\dot{\psi}_s = f_s$ with $s = 1,\ldots,\mathcal{S}+\mathcal{R}$ and its rate function $f_s(\psi,t)$ can be written in the same form as the original version Equation 2.10 and 2.11, if I set the stoichiometric coefficients

$$h_{s,r} = \delta_{s,\mathcal{S}+r} \quad \text{and} \quad g_{s,r} = 0, \quad \text{for} \begin{cases} s &= \mathcal{S}+1,\ldots,\mathcal{S}+\mathcal{R} \\ r &= 1,\ldots,\mathcal{R} \end{cases} \tag{4.8}$$

i.e., for all fictitious species, there is no net production and a fictitious species $s = \mathcal{S}+r$ affects only the rate of its own reaction r.

In the corresponding super-ensemble formulation $H = H(\mathbf{Y})$ and $\bar{Q} = \bar{Q}(\mathbf{Y})$, with expanded $\mathbf{Y}$-vector including $y_{i,s}$-variables for $s = 1,\ldots,\mathcal{S}+\mathcal{R}$, H_{K} and H_{X} can then be written as sums of local coupling terms, $h_{\mathrm{K}}^{(i,s)}(\mathbf{Y})$ and $h_{\mathrm{X}}^{(i,s)}(\mathbf{Y})$, respectively, which are local both in the time (i) and in the species (s) domain, as follows:

$$H_{\mathrm{K}}(\mathbf{Y}) = \sum_{i=0}^{\mathcal{I}}\sum_{s=1}^{\mathcal{S}+\mathcal{R}} h_{\mathrm{K}}^{(i,s)}(\mathbf{Y}) \tag{4.9}$$

$$H_{\mathrm{X}}(\mathbf{Y}) = \sum_{i=0}^{\mathcal{I}}\sum_{s=1}^{\mathcal{S}+\mathcal{R}} h_{\mathrm{X}}^{(i,s)}(\mathbf{Y}) \tag{4.10}$$

where $h_{\mathrm{K}}^{(i,s)}(\mathbf{Y})$ and $h_{\mathrm{X}}^{(i,s)}(\mathbf{Y})$ comprise all squared residues $\rho_{k,s}^{(\mathrm{K})}$ and $\rho_{n,s}^{(\mathrm{X})}$ whose time points at $\hat{t}_k$ or $\bar{t}_n$, respectively, fall within the interval

$$I_i = (t_{i-1}, t_i] = \{t | t_{i-1} < t \le t_i\} \tag{4.11}$$

So,

$$h_{\mathrm{K}}^{(i,s)} = \sum_{k,\hat{t}_k \in I_i} (\rho_{k,s}^{(\mathrm{K})})^2 \tag{4.12}$$

$$h_{\mathrm{X}}^{(i,s)} = \sum_{n,\bar{t}_n \in I_i} (\rho_{n,s}^{(\mathrm{X})})^2 \tag{4.13}$$

Futhermore, $h_{\mathrm{K}}^{(i,s)}$ and $h_{\mathrm{X}}^{(i,s)}$ depend only a local subset of variables $y_{i',s'}$ with (i', s') in the coupling neighborhood of (i, s), denoted by $\mathcal{V}(i, s)$, defined by

$$\mathcal{V}(i,s) = \mathcal{V}^{\mathcal{I}}(i) \times \mathcal{V}^{\mathcal{S}+\mathcal{R}}(s) = \{(i', s') \mid i' \in \mathcal{V}^{\mathcal{I}}(i), s' \in \mathcal{V}^{\mathcal{S}+\mathcal{R}}(s)\} \tag{4.14}$$

where

$$\begin{aligned} \mathcal{V}^{\mathcal{I}}(i) &= \{i' \mid 0 \leq i' \leq \mathcal{I}, \text{ and } i - i_\Phi \leq i' < i + i_\Phi\} & (4.15) \\ \mathcal{V}^{\mathcal{S}+\mathcal{R}}(s) &= \{s' \mid \; \leq s' \leq \mathcal{S} + \mathcal{R}, \exists r = 1, \ldots, \mathcal{R} : g_{sr} h_{rs'} \neq 0\} & (4.16) \end{aligned}$$

and i_Φ denotes the range of the FE basis functions $\Phi_i(t)$, i.e.,

$$i_\Phi = \begin{cases} 1 & \text{for the } 1^{st} \text{ order Lagrange and } 3^{rd} \text{ oder Hermite FE} \\ 2 & \text{for the } 2^{nd} \text{ order Lagrange FE} \end{cases} \tag{4.17}$$

Here $\mathcal{V}^{\mathcal{I}}(i)$ is the set of local temporal neighbors i' which are coupled to i within the finite range of the FE basis function $\Phi_i(t)$. Also $\mathcal{V}^{\mathcal{S}+\mathcal{R}}(s)$ is the set of all local species neighbors s' which are coupled to s by affecting the rate of any reaction r that either produces $(g_{s,r} > 0)$ or consumes $(g_{s,r} < 0)$ species s.

Due to the locality of $H(\mathbf{Y})$, a local update of a $y_{i's'}$ variable at site (i', s'):

$$y_{i',s'} \rightarrow y'_{i',s'} = y_{i',s'} + \Delta_{i',s'}(2u - 1), \tag{4.18}$$

with uniform random number $u \in [0, 1]$, changes the local coupling terms

$$h_{i,s}(\mathbf{Y}) = \beta_{\mathrm{K}} h_{\mathrm{K}}^{(i,s)}(\mathbf{Y}) + \beta_{\mathrm{X}} h_{\mathrm{X}}^{(i,s)}(\mathbf{Y}) \tag{4.19}$$

only for a small number of affected sites (i, s), i.e.,

$$\Delta H(\mathbf{Y} \rightarrow \mathbf{Y}') = H(\mathbf{Y}') - H(\mathbf{Y}) = \sum_{i,s \in \mathcal{U}(i',s')} [h_{i,s}(\mathbf{Y}') - h_{i,s}(\mathbf{Y})] \tag{4.20}$$

Here $\mathcal{U}(i', s')$ is the local "sphere of influence" of site (i', s'), defined in terms of the coupling neighbors $\mathcal{V}(i, s)$ (Equation 4.14) as follows:

$$\mathcal{U}(i', s') = \{(i, s) \quad | \quad (i', s') \in \mathcal{V}(i, s)\} \tag{4.21}$$

Note that, in general, $\mathcal{U}(i', s')$ is *not* the same as $\mathcal{V}(i', s')$, since the "connectivity relation" between s' and s,

$$\mathcal{C}(s, s') : \exists r \in \{1, \ldots, \mathcal{R}\} : g_{sr} h_{s'r} > 0 \tag{4.22}$$

of the (expanded) ODE network is generally *not* symmetric: species s' may affect the rate f_s for species s without $f_{s'}$ being affected by s. For example, in a single-step catalytic reaction

$$A + C \to B + C \tag{4.23}$$

the catalyst C affects the rate of net production of A and B, but A and B do *not* affect the production of C.

The locality of $\Delta H(\mathbf{Y} \to \mathbf{Y}')$ can be exploited to achieve an efficient scalable parallelization of the super-ensemble MC algorithm. This will now be described.

4.2 Parallel super-ensemble MC Algorithm

The MC updating procedure is organized into MC sweeps where one MC sweep consists of

$$\mathcal{D} = (\mathcal{I} + 1) \times (\mathcal{S} + \mathcal{R}) = \dim(\mathbf{Y}) \tag{4.24}$$

single-$y_{i,s}$ Metropolis moves and $\mathcal{D}$ is the dimension of $\mathbf{Y}$ vector, i.e,. the total number of $y_{i,s}$-variables. So during each sweep, each $y_{i,s}$ variable is visited once, on average, for a Metropolis move described in Section 3.1, Equation 3.9 and in Equation 4.18.

To distribute efficiently the task of performing such an MC sweep over multiple processors, I use a time-domain decomposition. Given $\mathcal{P}$ processors, numbered

$$p = 0, \ldots, \mathcal{P} - 1 \tag{4.25}$$

I assign a sub-domain of consecutive time slices i

$$i^{(-)}(p) \leq i \leq i^{(+)}(p) \tag{4.26}$$

to each processor p, as illustrated in Figure 4.1 for $\mathcal{I} = 15$, and $\mathcal{P} = 4$. Neighboring processors p and $p + 1$ control neighboring sub-domains, that is

$$i^{(-)}(p + 1) = i^{(+)}(p) + 1 \quad \text{for } p = 0, \ldots, \mathcal{P} - 2 \tag{4.27}$$

and the sub-domains cover all time slices:

$$i^{(-)}(0) = 0, \quad i^{(+)}(\mathcal{P} - 1) = \mathcal{I} \tag{4.28}$$

The size of the sub-domain of each processor

$$\Delta i(p) = i^{(+)}(p) - i^{(-)}(p) + 1 \tag{4.29}$$

must cover at least twice the range of the FE basis function Φ_i. That is, we require

$$\Delta i(p) \geq 2 i_\Phi \tag{4.30}$$

To achieve load balancing, the sub-domain sizes $\Delta i(p)$ of all processors should be approximately the same. So, ideally, if $\mathcal{I} + 1$ is divisible by $\mathcal{P}$,

$$\Delta i(p) = (\mathcal{I} + 1)/\mathcal{P} \quad \text{for all } p = 0, \ldots, \mathcal{P} - 1 \tag{4.31}$$

or else

$$\frac{\mathcal{I} + 1}{\mathcal{P}} - 1 < \Delta i(p) < \frac{\mathcal{I} + 1}{\mathcal{P}} + 1 \tag{4.32}$$

I expect linear speed-up of the computation as long as

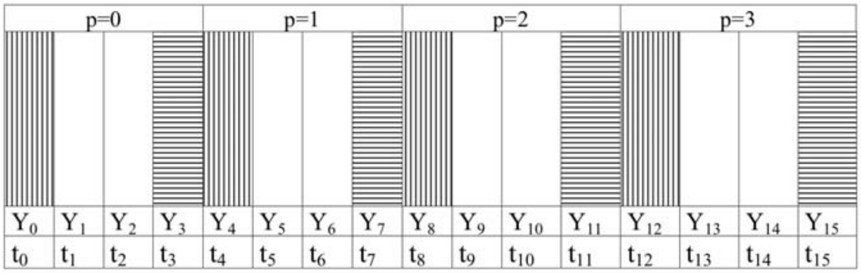
$$\Delta i(p) \gg 1 \tag{4.33}$$

or equivalently

$$\mathcal{I} + 1 \gg \mathcal{P} \tag{4.34}$$

When $\mathcal{P}$ becomes comparable to $\mathcal{I}$, the computation speed will begin to saturate, or even decrease, as interprocessor communication time, as described below, begin to dominate over intraprocessor computation.

For very large networks and correspondingly large $\mathcal{P}$, additional speed-up may be obtained by sub-dividing not only the time (i) but also the species (s) domain into sub-domains and exploiting the locality of $H(\mathbf{Y})$ in the s-domain. The s-domain decomposition should be constructed so that (i) the boundary "surface" of each s-sub-domain (i.e., the



p=0				p=1				p=2				p=3			
▥			▤	▥			▤	▥			▤	▥			▤
Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}

Figure 4.1: Illustration for time sub-domain layout for parallel updating sweep with $\mathcal{I} = 15$ and $\mathcal{P} = 4$. The sub-domain size of each processor is therefore $\Delta i(p) = 4$ with $p = 0, \ldots, \mathcal{P} - 1 = 3$. The vertically shaded columns denote $B^{(-)}(p)$, and the horizontally shaded columns denote $B^{(+)}(p)$, for $i_\Phi = 1$, with $p = 0, \ldots, \mathcal{P} - 1 = 3$. Each $\mathbf{Y}_i$ for $i = 0, \ldots, \mathcal{I} = 15$ is a $(\mathcal{R} + \mathcal{S})$- dimensional vector: $\mathbf{Y}_i = (y_{i,s}|s = 1, \ldots, \mathcal{R} + \mathcal{S})$.

subset of sites s that are connected to different sub-domain) is minimized; and (ii) all s-sub-domains are of approximately equal size. In the present work, I will not pursue such a species domain composition, but rather consider only time domain decomposition.

In each MC sweep, each processor p performs

$$\Delta i(p) \times (\mathcal{S} + \mathcal{R}) \tag{4.35}$$

Metropolis updates on the $y_{i,s}$-variables of the (i, s)-site of the processors time-species sub-domain, with $i^{(-)}(p) \leq i \leq i^{(+)}(p)$, and $1 \leq s \leq \mathcal{S} + \mathcal{R}$. The random selection of the update site (i, s) during such a sweep is constrained by the requirement that neighboring processors p_1 and $p_2 = p + 1$ must *not* simultaneously perform updates on time slices i_1 and i_2, respectively, which fall within range i_Φ from the common boundary of p_1 and p_2. If i_1 and i_2 do fall within this boundary region, they are connected across the sub-boundary by a coupling term $h(i_1, s_1)$ or $h(i_2, s_2)$ which can depend both on y_{i_1,s_1} and on y_{i_2,s_2}; so p_1 cannot update y_{i_1,s_1}, unless y_{i_2,s_2} is kept constant by p_2 and vice versa. These boundaries layers for processor p are defined by

$$\begin{aligned} B^{+}(p) &= \{(i,s) &\mid& \quad 1 \leq s \leq \mathcal{S}+\mathcal{R}, i^{(+)}(p) - i_{\Phi} + 1 \leq i \leq i^{(+)}(p)\} \qquad (4.36) \\ \text{and } B^{(-)}(p) &= \{(i,s) &\mid& \quad 1 \leq s \leq \mathcal{S}+\mathcal{R}, i_{(-)}(p) \leq i \leq i^{(-)}(p) + i_{\Phi} - 1\} \qquad (4.37) \end{aligned}$$

Note that these two boundary layers are non-overlapping,

$$B^{(+)}(p) \cap B^{(-)}(p) = \oslash \qquad (4.38)$$

as long as the condition Equation 4.30 is obeyed.

To enforce the sub-domain boundary layer constraint, all processors p are embargoed from visiting any site $(i,s) \in B^{(-)}(p)$ for updating $y_{i,s}$ during MC sweeps with odd sweep number l; they are embargoed from visiting any sites $(i,s) \in B^{(+)}(p)$ during MC sweeps with even MC sweep number l. For each single-$y_{i,s}$ update, each processor first selects a random site (i,s) with uniform probability from its sub-domains, without regards to the embargo. If (i,s) does not fall within the embargoed boundary layer, the Metropolis update is performed on $y_{i,s}$. However, if (i,s) *does* fall within the embargoed boundary layer, (i,s) is replaced by its "mirror image", $(\bar{i},s)$ in the opposite layer, i.e.,

$$\bar{i} = i^{(+)}(p) + i^{(-)}(p) - i \qquad (4.39)$$

Instead of $y_{i,s}$, the variable $y_{\bar{i},s}$ is then subjected to the Metropolis update.

By enforcing the embargo in this manner, each site (i,s) will, on average over many sweeps, receive the same number of updating hits. That is, a site (i,s) outside the boundary layers will be visited during each MC update with probability:

$$P_{visited}(i,s) = \frac{1}{\Delta i(p)(\mathcal{S}+\mathcal{R})} \quad \text{if } (i,s) \notin [B^{(+)}(p) \cup B^{(-)}(p)], \qquad (4.40)$$

whereas sites (i,s) inside either boundary layer will be visited with probability

$$P_{visited}(i,s) = \begin{cases} \frac{2}{\Delta i(p)(\mathcal{S}+\mathcal{R})} & \text{if } (i,s) \in B^{(+)}(p),\ l \text{ is odd, or } (i,s) \in B^{(-)}(p),\ l \text{ is even} \\ 0 & \text{otherwise.} \end{cases} \qquad (4.41)$$

In order to perform updates of sites (i,s) in either boundary layer $B^{\pm}(p)$, processor p must also have access to the y-variables of the adjacent boundary $B^{\mp}(p \pm 1)$ of the neighbor processor. That is to update an $(i,s) \in B^{\pm}(p)$, processor p needs to access the values of $y_{i',s'}$ for $(i',s') \in B^{\mp}(p \pm 1)$. Therefore, each processor must communicate the up-to-date $y_{i,s}$-values of its $B^{(+)}(p)$ $(B^{(-)}(p))$ to its right (left) neighbor $p+1$ $(p-1)$ after completing an odd-(even-) numbered MC sweep.

To ensure that each processor can gain access to the up-to-date y-variables of the adjacent boundary $B^{\mp}(p \pm 1)$ of the neighbor processor, I also have to set a barrier after completion of each MC sweep. That is to say, if one processor finishes one MC sweep, it should not continue to the next MC sweep, instead, it has to wait until other processors finish that MC sweep. Therefore, the processor $p = 0$ also serves as “a master processor” during the simulation to keep track of the MC updating sweep number l and to keep all other processor properly synchronized after completion of each MC sweep.

Chapter 5

Virtual Experimental Data and Monte Carlo Protocols

5.1 The Enzyme Model as a Virtual Test Case

As a simple test problem, I have applied the super-ensemble approach to the enzyme model described in Section 2.1, Figure 2.1. I am using "virtual" experimental data as input to the ensemble simulation. These virtual data are generated by solving the ODE model (Equation 2.5 to 2.8) for a given set of "true" model parameters $\mathbf{\Gamma}^{(\mathrm{true})}$ listed in Table 5.1(a). This "true" ODE solution was obtained by a standard numerical ODE solver using second order BDF(backward differentiation formula) ODE integration method. This "true" solution has a numerical accuracy of 1 part in 10^8 or better.

A set of $\mathcal{N} = 22$ "observation" time points $\bar{t}_n$ between $t_0 = 0$ and $t_{\mathcal{I}} = 8$ model time unit were randomly chosen, as shown in Table 5.2. For each of the $\mathcal{S} = 4$ species in the model (s $= S$, E, ES_2, or P) I select a subset of these $\bar{t}_n$ to generate "virtual" data points $Z_{n,s}$ by adding a certain amount of Gaussian random noise to the true ODE solution $\psi_s^{(\mathrm{true})}(t)$,i.e.,

$$Z_{n,s} = \psi_s^{(\mathrm{true})}(\bar{t}_n) + \Delta Z_{n,s} \tag{5.1}$$

where

$$\psi_s^{(\mathrm{true})}(t) := \psi_s(t; \mathbf{\Gamma}^{(\mathrm{true})}) \tag{5.2}$$

and $\Delta Z_{n,s}$ is drawn from a Gaussian distribtution

$$p_{n,s}(\Delta Z) = \frac{1}{\sqrt{2\pi}\sigma_{n,s}} \exp[-(\Delta Z)^2/(2\sigma_{n,s}^2)] \tag{5.3}$$

with a standard deviation $\sigma_{n,s}$, given in terms of a reference concentration Z_{ref}

$$\sigma_{n,s} = \omega_\sigma \times Z_{ref} \tag{5.4}$$

Table 5.1: Parameters in Enzyme Model

(a) List of all the “true” values of seven model parameters $\Gamma^{(\text{true})}$ used in generating ODE solutions by standard numerical ODE solver and their corresponding lower $y_{i,s}^{(\text{lo})}$ and higher $y_{i,s}^{(\text{hi})}$ boundaries in super-ensemble simulation

Model Parameter	$\Gamma^{(\text{true})}$	$y_{i,s}^{(\text{lo})}$	$y_{i,s}^{(\text{hi})}$
$[E](t_0)$	2.400000	0.000001	1,000
$[ES_2](t_0)$	0.000030	0.000001	1,000
$[P](t_0)$	0.000020	0.000001	1,000
$[S](t_0)$	26.00000	0.000001	1,000
θ_1	0.000960	0.00001	10
θ_2	0.102000	0.00001	10
θ_3	0.190000	0.00001	10
θ_4	0	0	0

(b) List of all control parameter in super-ensemble simulation

Number of Species	$\mathcal{S}$	4
Number of Rate Coefficients	$\mathcal{R}$	3
Number of Experimental Data	$\mathcal{N}$	22
Number of interpolation grid points t_i	$\mathcal{I}+1$	16, 32, 64
Number of check points $\hat{t}_k$	$\mathcal{K}$	15, 31, 63

Here, Z_{ref} is the maximum "true" initial concentration of all species,

$$Z_{ref} = \max_{s}(\psi_s^{(\text{true})}(t = t_0)) \tag{5.5}$$

i.e., from the "true" parameter values in Table 5.1(a), Z_{ref} is the initial concentration of "true" species S, $Z_{ref} = 26$ model concentration units.

Table 5.2: List of all three sets of virtual experimental data set generated from the true kinetics solution, with $\omega_\sigma = 0\%$, 1%, or 2%, respectively

Index n	Observation Time $\bar{t}_n$	Species s	Data Set 0 $\omega_\sigma = 0\%$ $Z_{n,s}$	Data Set 1 $\omega_\sigma = 1\%$ $Z_{n,s}$	Data Set 2 $\omega_\sigma = 2\%$ $Z_{n,s}$
1	0.7	P	0.0318	0.235	0.503
2	0.9	E	1.492	1.726	1.961
3	1.4	E	1.256	1.374	1.492
4	1.5	P	0.113	0.0214	0.156
5	1.6	S	23.330	23.624	23.918
6	1.8	ES_2	1.263	1.572	1.882
7	2.5	ES_2	1.385	1.121	0.858
8	3.3	E	0.949	1.003	1.058
9	3.4	P	0.376	0.518	0.659
10	3.9	ES_2	1.472	1.458	1.444
11	4.2	P	0.496	0.603	0.710
12	4.3	P	0.511	0.486	0.460
13	4.4	E	0.918	0.894	0.871
14	4.9	S	21.825	21.232	20.638
15	5.3	E	0.915	1.040	1.166
16	6.4	S	21.386	22.123	22.860
17	6.6	E	0.924	1.274	1.623
18	6.6	ES_2	1.475	1.680	1.886
19	6.9	E	0.927	1.561	2.195
20	7.1	P	0.934	1.064	1.194
21	7.6	E	0.935	1.031	1.128
22	7.7	S	21.026	20.846	20.666

There are three such data set shown in Table 5.2, generated with $\omega_\sigma = 0\%$ (Data Set 0), $\omega_\sigma = 1\%$ (Data Set 1) and $\omega_\sigma = 2\%$ (Data Set 2). So Data Set 0 consists simply of the "true" ODE solution $\psi_s^{(\text{true})}(\bar{t}_n)$ without any noise added.

If, for a certain species s and time point $\bar{t}_n$, no virtual observation $Z_{n,s}$ was made, I set formally $\sigma_{n,s} = \infty$ in H_{X} for such a data point, i.e., no data point is included for that (s, n)-combination in Equation 3.14. On the other hand, if I decide that an observation $Z_{n,s}$ was made at $\bar{t}_n$, then I set $\sigma_{n,s}$ in Equation 3.14 to $\sigma_{n,s} = \omega_\sigma \times Z_{ref}$ and draw a random $\Delta Z_{n,s}$ to generate $Z_{n,s}$ for Data Sets 1 and 2 . Also I set in H_{X}

$$\sigma_{n,s} = 1\% \times Z_{ref} \tag{5.6}$$

for the zero-noise Data Set 0 so as to obtain a well defined continuous ensemble distribution for this case, even though the actual data points $Z_{n,s}$ were generated with $\sigma_{n,s} = 0$

5.2 Monte Carlo Protocol: Initialization, Annealing, Equilibration and Accumulation

5.2.1 MC Initialization Phase

I am using the expanded ODE model formulation described in Section 4.1 where the rate coefficient variables θ_r are replaced by fictitious species $\Psi_{\mathcal{S}+r}(t)$. Therefore, in the super-ensemble simulation all MC degrees of freedom are $y_{i,s}$- variables with $s = 1, \ldots, \mathcal{S} + \mathcal{R}$ and $\mathcal{S} = 4, \mathcal{R} = 3$.

In the super-ensemble simulation, I impose lower and upper limits on all $\mathbf{Y}$-variables indicated by the columns labeled *lo* and *hi* in Table 5.1(a). If a new $y'_{i,s}$-value proposed during a Metropolis update falls outside of these $[y_{i,s}^{(\text{lo})}, y_{i,s}^{(\text{hi})}]$ intervals, the proposed move is automatically rejected.

Each simulation starts from a completely randomly chosen initial configuration $\mathbf{Y}^{(\text{init})}$ within $[lo, hi]$ interval, i.e.

$$y_{i,s}^{(\text{init})} = y_{i,s}^{(\text{lo})} + u \times (y_{i,s}^{(\text{hi})} - y_{i,s}^{(\text{lo})}) \tag{5.7}$$

where u is a uniform random number in $[0, 1]$. All random numbers u in the simulations reported here were obtained by the RANECU pseudo random number generator algorithm [7] with a period of approximately 10^{18}.

5.2.2 MC Annealing Phase

In order to find the region in $\mathbf{Y}$-space where H_{K} and H_{X} are close to minimal, I then perform a Monte Carlo annealing phase [9], consisting of

$$\bar{L}_{\mathrm{An}} = 1,000,000\text{MC Sweeps} \tag{5.8}$$

During this annealing phase, both β_{X} and β_{K} in Equation 3.13 are gradually increased from some initial value to a larger final value,

$$\beta_\alpha^{(\mathrm{init})} \longrightarrow \beta_\alpha^{(\mathrm{finl})} \quad \text{for } \alpha = \mathrm{X}, \mathrm{K}, \tag{5.9}$$

according to the following annealing schedule

$$\beta_\alpha(l) = \beta_\alpha^{(\mathrm{init})} \times (\beta_\alpha^{(\mathrm{finl})} / \beta_\alpha^{(\mathrm{init})})^{(l/L_{\mathrm{An}})} \quad \text{for } \alpha = \mathrm{X}, \mathrm{K} \tag{5.10}$$

where $l = 0, \ldots, L_{\mathrm{An}} - 1$ and

$$L_{\mathrm{An}} = 1,000 \tag{5.11}$$

is the number of “annealing periods” and each annealing period l comprises

$$\frac{\bar{L}_{\mathrm{An}}}{L_{\mathrm{An}}} = 1,000 \frac{\text{MC Sweeps}}{\text{Annealing Period}} \tag{5.12}$$

During each annealing period l, i.e., for MC sweeps $\bar{l}$ with

$$l \times L_{\mathrm{An}} \leq \bar{l} < (l+1) \times L_{\mathrm{An}} \tag{5.13}$$

the values of β_{X} and β_{K} are kept constant.

For the results reported below in Chapter 6 I used

$$\beta_{\mathrm{X}}^{(\mathrm{init})} = 1.352 \tag{5.14}$$

$$\beta_{\mathrm{X}}^{(\mathrm{finl})} = 3.95 \times 10^6 \tag{5.15}$$

$$\beta_{\mathrm{K}}^{(\mathrm{init})} = 1 \tag{5.16}$$

$$\beta_{\mathrm{K}}^{(\mathrm{finl})} = 3.98 \times 10^8 \tag{5.17}$$

5.2.3 MC Equilibration Phase

The annealing phase is then followed by an equilibration phase, of duration

$$\bar{L}_{\mathrm{Eq}} = 8,000,000 \quad \text{MC Sweeps} \tag{5.18}$$

During this phase, β_{K} is either kept at the $\beta_{\mathrm{K}}^{(\mathrm{finl})}$-value reached during the annealing phase, or β_{K} is dropped to a constant lower value $\beta_{\mathrm{K}}^{(\mathrm{Eq})} < \beta_{\mathrm{K}}^{(\mathrm{finl})}$. Also β_{X} is dropped to its constant "target" value, i.e., to

$$\beta_{\mathrm{X}}^{(\mathrm{Eq})} = 1 \tag{5.19}$$

In this manner, the MC Markov chain is allowed to equilibrate to the actual "target" ensemble distribution

$$\bar{Q}(\mathbf{Y}) = \frac{1}{\mathcal{Z}} \exp[-(\beta_{\mathrm{X}}^{(\mathrm{Eq})} H_{\mathrm{X}} + \beta_{\mathrm{K}}^{(\mathrm{Eq})} H_{\mathrm{K}})] \tag{5.20}$$

with $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ kept at a large enough value so that the variational solutions $\boldsymbol{\Psi}(t;\mathbf{Y})$ are still reasonably accurate solutions to the ODE system. At the same time, $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ should be low enough to still permit efficient equilibration of the MC Markov process.

5.2.4 MC Accumulation Phase

The equilibration phase is followed by an accumulation phase of duration

$$\bar{L}_{\mathrm{Ac}} = 3,000,000 \quad \text{MC Sweeps} \tag{5.21}$$

with the same weights $\beta_{\mathrm{X}}^{(\mathrm{Eq})} = 1$ and $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ as in the equilibration phase. During this accumulation phase, a total number of

$$L_{\mathrm{Ac}} = 30,000 \tag{5.22}$$

MC sample points $\mathbf{Y}^{(l)}$ are collected, for $l = 1, \ldots, L_{\mathrm{Ac}}$, with a sampling period of

$$\frac{\bar{L}_{\mathrm{Ac}}}{L_{\mathrm{Ac}}} = 100 \frac{\text{MC Sweeps}}{\text{Sample Point}} \tag{5.23}$$

So the sample points, denoted by

$$\mathbf{Y}^{(l)} = (\ldots, y_{i,s}^{(l)}, \ldots) \tag{5.24}$$

are collected after every 100 MC sweeps. Each sample point and several other quantities depending on $\mathbf{Y}^{(l)}$ are written out into a sample output file for further statistical analysis after completion of the ensemble simulation.

5.3 MC Error Estimation

The foregoing four-phase simulation procedure consisting of MC random initialization, annealing, equilibration and accumulation is repeated

$$\mathcal{J} = 10 \tag{5.25}$$

times. For each such repeat, $j = 1, \ldots, \mathcal{J}$, a different new MC inital $\mathbf{Y}^{(\mathrm{init})}$ is generated according to Equation 5.7. These repeats are used to estimate the MC error bars of the simulation as follows: let $\mathbf{Y}^{(l,j)}$ denote the l^{th} sample point collected during the j^{th} repeat. Then the "partial" MC average of some quantity $A(\mathbf{Y})$ for the j^{th} repeat is given by

$$< A(.) >^{(j)} = \frac{1}{L_{\mathrm{Ac}}} \sum_{l=1}^{L_{\mathrm{Ac}}} A(\mathbf{Y}^{(l,j)}) \tag{5.26}$$

and its overall MC average

$$< A(.) >_{\mathrm{MC}} = \frac{1}{\mathcal{J}} \sum_{j=1}^{\mathcal{J}} < A(.) >^{(j)} \tag{5.27}$$

This MC average should approach the ensemble average $< A >$ defined in Equation 3.7 in the limit $L_{\mathrm{Ac}} \to \infty$, according to the Central Limit Theorem.

From the statistical "spread" of these partial MC average I can then estimate the MC standard deviation(MCSD) of $A(\mathbf{Y})$ by:

$$\sigma_{\mathrm{MC}}[A] = [\frac{1}{\mathcal{J}-1}(\sum_{j=1}^{\mathcal{J}}(< A(.) >^{(j)})^2 - \mathcal{J}(\frac{1}{\mathcal{J}}\sum_{j=1}^{\mathcal{J}} < A(.) >^{(j)})^2)]^{1/2} \tag{5.28}$$

This MCSD characterizes the statistical convergence of the MC sampling procedure and it depends on the total MC sample size $L_{\mathrm{Ac}} \times \mathcal{J}$ such that

$$\sigma_{\mathrm{MC}}[A] \simeq \frac{1}{\sqrt{L_{\mathrm{Ac}}\mathcal{J}}} \tag{5.29}$$

for sufficient large L_{Ac}. So $\sigma_{\mathrm{MC}}[A]$ should vanish for $L_{\mathrm{Ac}} \to \infty$.

The MC standard deviation should be clearly distinguished from the *ensemble* standard deviation (ESD) $\sigma[A]$ defined in Section 3.1, Equation 3.8. Clearly $\sigma[A]$ is a quantity that depends only on the ensemble distribution $\bar{Q}(\mathbf{Y})$; $\sigma[A]$ should *not* depend on any MC sampling procedure or on the MC sample size $L_{\mathrm{Ac}}\mathcal{J}$ if the MC simulation is properly equilibrated. That is, for $L_{\mathrm{Ac}} \to \infty$, $\sigma[A]$ calculated by MC simulation, should approach the sample-size-independent value given by Equation 3.8. $\sigma[A]$ characterizes how well (or poorly) the available experimental data included in H_{X} constrain the ensemble prediction for the outcome of a future measurement of the quantity $A(\mathbf{Y})$. Therefore, it enables us to gain insights into what are likely to be the most informative new experiments that should be done to reduce the uncertainty in the model predictions.

5.4 MC Stepwidth Optimization

The allowable range of original model parameter variables (θ_r, X_s) can cover many order of magnitude in typical biological network simulations. One must account for this in performing Metropolis updating steps with proposed moves of the $\mathbf{Y}$-variables

$$y_{i,s} \to y'_{i,s} = y_{i,s} + \Delta_{i,s}(2u-1) \tag{5.30}$$

where u is a uniform random number drawn form $[0,1]$. Namely, I must choose updating stepwidths $\Delta_{i,s}$ that are compatible, in order of magnitude, with the typical range of $y_{i,s}$-values being sampled in $\bar{Q}(\mathbf{Y})$ during the MC Markov chain process. If we set the $\Delta_{i,s}$-value to be much larger than the typical $y_{i,s}$ -range, most proposed $y'_{i,s}$ will result in a very large energy $H(\mathbf{Y}')$. Most such "large" moves would be rejected and, as a result, the Markov chain process would equilibrate very slowly. On the other hand, if we set $\Delta_{i,s}$ much smaller than the typical $y_{i,s}$-range, the proposed energy change $\Delta H(\mathbf{Y} \to \mathbf{Y}')$ will be very small and all such moves very likely be accepted. However the MC Markov chain will again equilibrate very slowly since each such "small" move changes $y_{i,s}$ by "almost" negligible amount.

Unfortunately, at the beginning of the MC simulation, I do not know the optimal values of $\Delta_{i,s}$ since I have no knowledge at all about the typical $y_{i,s}$-variable ranges to be sampled. However, based on the foregoing considerations, I can use the Metropolis acceptance probability as an indicator of whether $\Delta_{i,s}$ is too small or too large: I should choose $\Delta_{i,s}$ so that the average Metropolis acceptance probability for a move $y_{i,s} \rightarrow y'_{i,s}$,

$$p_{i,s}(\mathbf{Y} \rightarrow \mathbf{Y}') = \min(1, \exp[-\Delta H(\mathbf{Y} \rightarrow \mathbf{Y}')]) \tag{5.31}$$

is neither too close to 100% nor too close to 0%. So, as a rule of thumb, I want to adjust $\Delta_{i,s}$ so that

$$< p_{i,s}(\mathbf{Y} \rightarrow \mathbf{Y}') > \simeq 50\% \tag{5.32}$$

To achieve this, I start the simulation with some initial guess $\Delta_{i,s}^{(\mathrm{init})}$, for example,

$$\Delta_{i,s}^{\mathrm{init}} = y_{i,s}^{(\mathrm{hi})} - y_{i,s}^{(\mathrm{lo})} \tag{5.33}$$

where $y_{i,s}^{(\mathrm{lo})}$ and $y_{i,s}^{(\mathrm{hi})}$ are the upper and lower limits imposed on $y_{i,s}$ during the simulation, as given in Table 5.1(a). During the entire simulation, I then repeatedly measure the acceptance probability for each variable $y_{i,s}$ over a certain number of MC sweeps by counting up the number of proposed moves of $y_{i,s} \rightarrow y'_{i,s}$, denoted by $N_{\mathrm{prop}}(i,s)$, and the number of those proposed moves which were actually accepted $N_{\mathrm{accp}}(i,s)$. The value of $N_{\mathrm{prop}}(i,s)$ is checked for all $y_{i,s}$ at the end of each MC sweep and when it reaches or exceeds a value of

$$N_{\mathrm{prop}}(i,s) = 20 \tag{5.34}$$

I do an "acceptance check", i.e., I estimate the average acceptance probability by taking the so-called "acceptance ratio"

$$p_{\mathrm{accp}}(i,s) = N_{\mathrm{accp}}(i,s)/N_{\mathrm{prop}}(i,s). \tag{5.35}$$

I would like to choose or adjust $\Delta_{i,s}$ so that $p_{\mathrm{accp}}(i,s)$ stays within some target range $[p_{\mathrm{accp}}^{(\mathrm{lo})}, p_{\mathrm{accp}}^{(\mathrm{hi})}]$ around 50%. In the simulation described below, I have chosen

$$p_{\mathrm{accp}}^{(\mathrm{lo})} = 35\%, \quad p_{\mathrm{accp}}^{(\mathrm{hi})} = 65\% \tag{5.36}$$

If the actual $p_{\mathrm{accp}}(i,s)$-value found in the acceptance check is within target range, I leave $\Delta_{i,s}$ unchanged; if $p_{\mathrm{accp}}(i,s)$ is too high(low), I increase(decrease) $\Delta_{i,s}$ as follow:

$$\Delta_{i,s} \to \Delta'_{i,s} = \begin{cases} \Delta_{i,s} \times q_{i,s} & \text{if } p_{\mathrm{accp}}(i,s) > p^{(\mathrm{hi})}_{\mathrm{accp}} \\ \Delta_{i,s} & \text{if } p_{\mathrm{accp}}(i,s) \in [p^{(\mathrm{lo})}_{\mathrm{accp}}, p^{(\mathrm{hi})}_{\mathrm{accp}}\ ] \\ \Delta_{i,s}/q_{i,s} & \text{if } p_{\mathrm{accp}}(i,s) < p^{(\mathrm{lo})}_{\mathrm{accp}} \end{cases} \tag{5.37}$$

Here, $q_{i,s}$ is the so-called stepwidth adjustment factor, with an appropriately chosen value

$$q_{i,s} > 1. \tag{5.38}$$

At the start of the MC simulation, $q_{i,s}$ is initialized to

$$q^{(\mathrm{init})}_{i,s} = 1.50 \tag{5.39}$$

During each acceptance check and before updating $\Delta_{i,s}$ according to Equation 5.37, the value of $q_{i,s}$ is also adjusted up or down, if needed. The $q_{i,s}$-adjustment depends on the current value of $p_{\mathrm{accp}}(i,s)$ and its value found in the most recent prior acceptance check, as follows:

If the current $p_{\mathrm{accp}}(i,s)$-value is within target range, leave $q_{i,s}$ unchanged. If $p_{\mathrm{accp}}(i,s)$ has remained above target range or has remained below target range for the current and last acceptance check , adjust $q_{i,s}$ upward, subject to an upper limit of

$$q^{(\mathrm{max})} = 5.0 \tag{5.40}$$

as follows:

$$q_{i,s} \to q'_{i,s} = \min(q^{(\mathrm{max})}, 1 + f^{(+)}_q(q_{i,s} - 1)) \tag{5.41}$$

where $f^{(+)}_q$ is set to

$$f^{(+)}_q = 1.25 \tag{5.42}$$

If, on the other hand, the most recent prior acceptance check has resulted in an “overshoot”, i.e., the most recent prior $p_{\mathrm{accp}}(i,s)$ was above and the current $p_{\mathrm{accp}}(i,s)$ is below target range, or vice versa, $q_{i,s}$ is adjusted downward, subject to a lower limit of

$$q^{(\mathrm{min})} = 1.05 \tag{5.43}$$

as follows

$$q_{i,s} \rightarrow q'_{i,s} = \max(q^{(\min)}, 1 + f_q^{(-)}(q_{i,s} - 1)) \tag{5.44}$$

where $f_q^{(-)}$ is set to

$$f_q^{(-)} = 0.75 \tag{5.45}$$

This adjusted value of $q'_{i,s}$ is then used to perform the update of $\Delta_{i,s}$ according to Equation 5.37. In the simulations reported below, we find that this algorithm successfully brings or restores p_{accp} into the prescribed target range with typically $5 - 10$ acceptance checks, corrsponding to 100-200 MC sweeps.

Chapter 6

Application to enzymatic network

6.1 Testing the Finite-Element Based Variational Approach

Before applying the FF-based Galerkin variational approach to super-ensemble Monte Carlo simulations, I should first test whether the FE basis and the variational method can indeed reproduce the ODE solutions for the model with sufficient accuracy. To do so, I first calculate a highly accurate reference solution $\psi_s^{(\text{true})}(t)$ for the "true" model parameter set $\mathbf{\Gamma}^{(\text{true})}$ listed in Table 5.1(a) on a sufficiently dense time grid with a relative numerical solution accuracy of 10^{-8} or better, over the simulation time interval $[t_0, t_{\mathcal{I}}] = [0, 8]$. This is done with second order backward differentiation formula (BDF) ODE integration method. I then initialize the $\mathbf{Y}$-vector by

$$y_{i,s}^{(\text{init})} := \begin{cases} \psi_s^{(\text{true})}(t_i) & \text{for } s = 1, \ldots, \mathcal{S} \\ \theta_r & \text{for } s = \mathcal{S} + r; r = 1 \ldots, \mathcal{R} \text{ and all i} \end{cases} \tag{6.1}$$

From $y_{i,s}^{(\text{init})}$, I calculate the corresponding FE approximand

$$\Psi_s^{(R)}(t) := \sum_{i=0}^{\mathcal{I}} y_{i,s}^{(\text{init})} \Phi_i(t) = \Psi_s(t; \mathbf{Y}^{(\text{init})}) \tag{6.2}$$

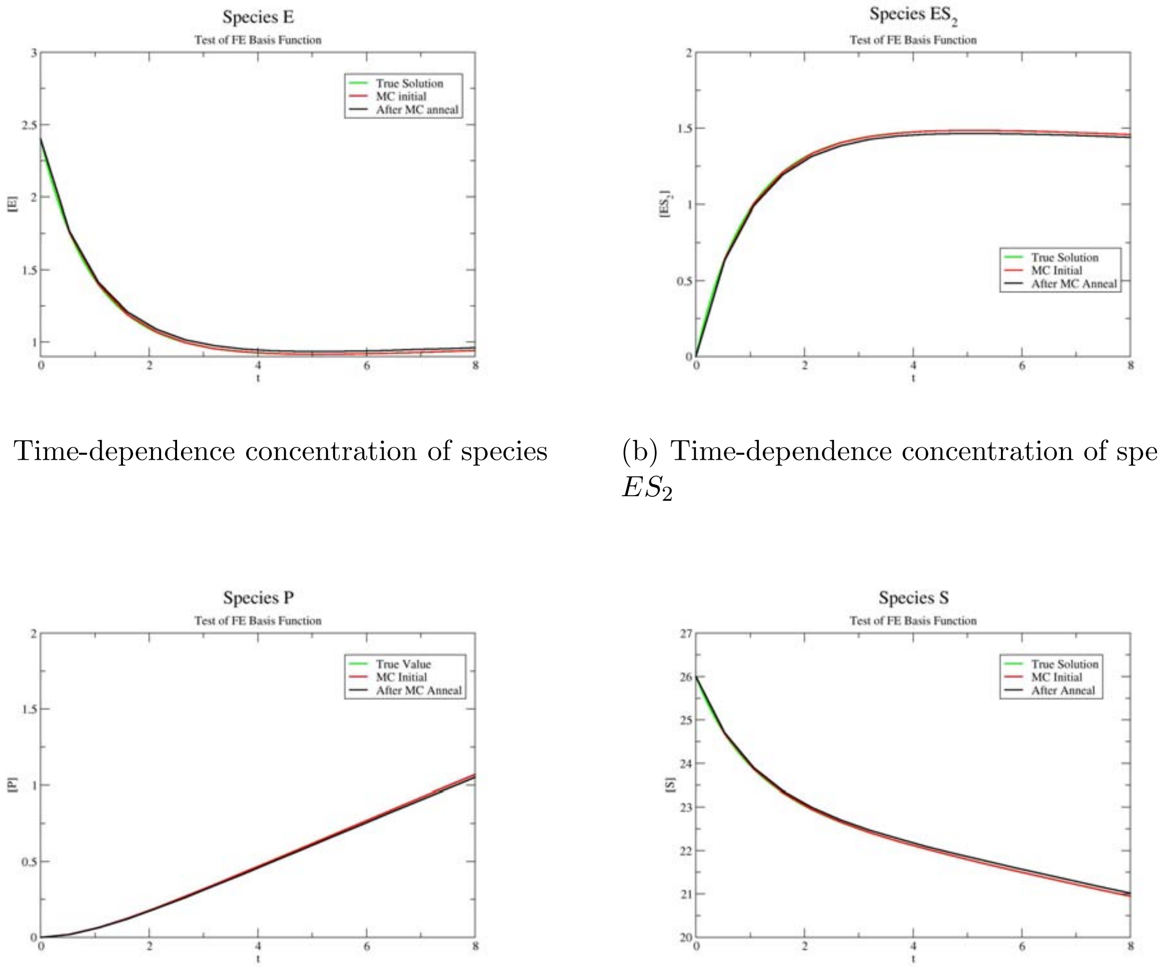
as well as its relative error, the so-called representation error defined as

$$E_s^{(R)}(t) = 2[\Psi_s^{(R)}(t) - \psi_s^{(\text{true})}(t)]/(||\Psi_s^{(R)}|| + ||\psi_s^{(\text{true})}||) \tag{6.3}$$

where, for any time-dependent function $\psi(t)$ I define a euclidean norm by

$$||\psi|| = [\frac{1}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} (\psi(\hat{t}_k))^2]^{1/2} \tag{6.4}$$

$E_s^{(R)}(t)$ provides a relative error measure of how well the true kinetics solution can be approximated in terms of an FE basis function expansion. As shown in Figures 6.1 and 6.2, already an FE basis of $\mathcal{I} = 15$ interpolation intervals, corresponding to $\mathcal{I} + 1 = 16$, the first order Lagrange FE basis is quite sufficient to obtain a $\Psi_s^{(R)}$ which approximates $\psi_s^{(\text{true})}$ to better than $||E_s^{(R)}(t)|| \leq 1.5\%$ accuracy. This level of accuracy should be quite sufficient for a super-ensemble simulation on typical noisy experimental data.



(a) Time-dependence concentration of species E

(b) Time-dependence concentration of species ES_2

(c) Time-dependence concentration of species P

(d) Time-dependence concentration of species S

Figure 6.1: The black lines represent MC initial $\Psi_s^{(R)}(t) = \Psi_s(t; \mathbf{Y}^{(\text{init})})$; the green lines represent the true kinetics solution $\psi_s^{(\text{true})}$; and the red lines represent variational solutions after MC annealing phase $\Psi_s^{(V)}(t) = \Psi_s(t; \mathbf{Y}^{(\text{finl})})$.

Next, I want to test whether the Galerkin variational approach is in fact capable of generating a reasonably accurate approximation to the true kinetics solution. To do so, with $\beta_{\mathrm{K}}^{(\mathrm{finl})} = 3.98 \times 10^8$, I perform a MC annealing calculation following the protocol described in Section 5.2.2 with the above $\mathbf{Y}^{(\mathrm{init})}$ (Equation 6.1) as the MC initialization. In H_{X} I include the time initial concentration values as the only virtual experimental data points, i.e.,

$$\begin{aligned} Z_{n,s} &= \psi_s^{(\mathrm{true})}(t_0) && (6.5) \\ \sigma_{n,s} &= 1\% \times Z_{ref} = 0.26 && (6.6) \\ \bar{t}_n &= t_0 && (6.7) \end{aligned}$$

So the initial condition $\psi_s^{(\mathrm{true})}(t_0)$ is the only virtual experimental data point for each real species s. From the final $\mathbf{Y}$-vector, $\mathbf{Y}^{(\mathrm{finl})}$, generated by this annealing procedure, I calculate the corresponding variational minimal-energy FE approximand:

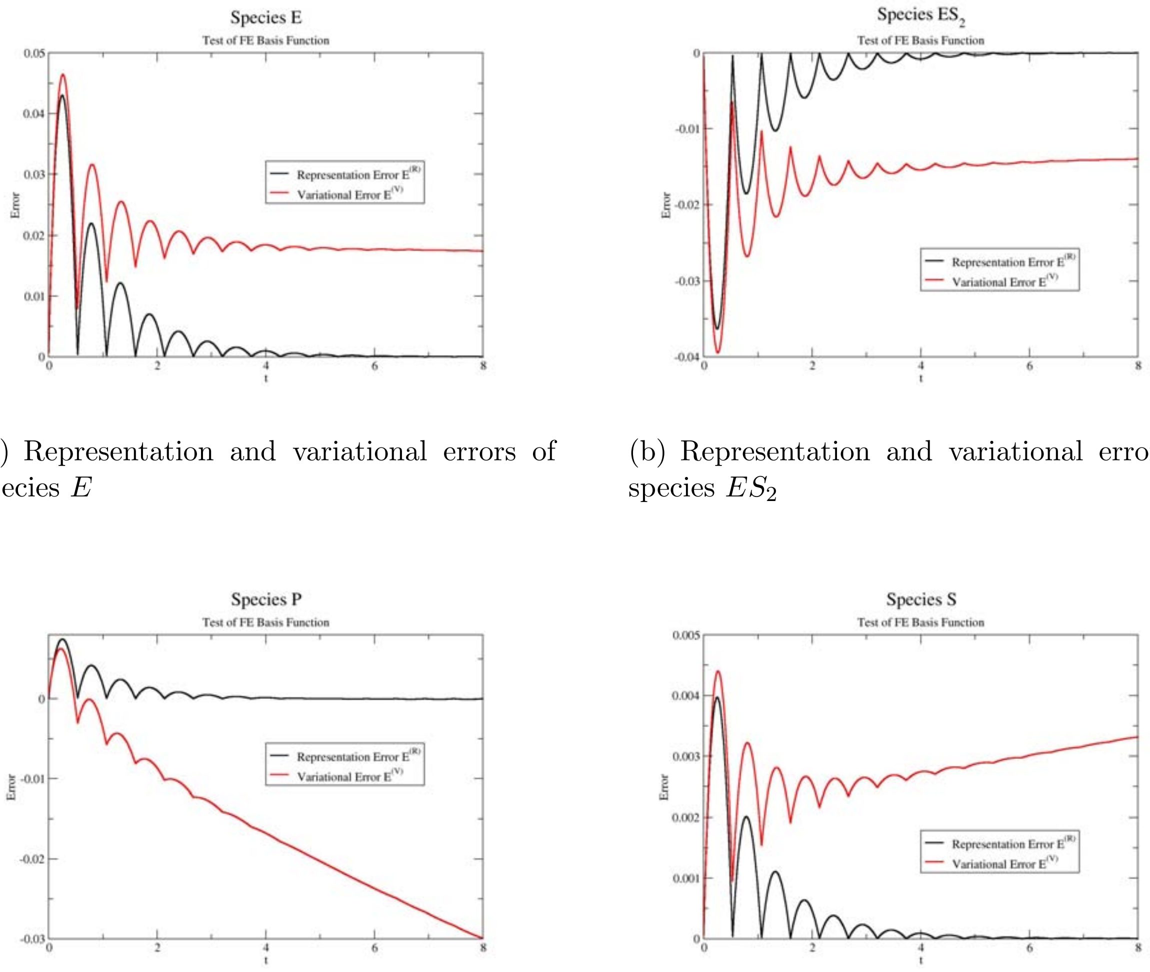
$$\Psi_s^{(V)}(t) := \sum_{i=0}^{\mathcal{I}} y_{i,s}^{(\mathrm{finl})} \Phi_i(t) = \Psi_s(t, \mathbf{Y}^{(\mathrm{finl})}) \qquad (6.8)$$

and its relative error, the so-called variational error:

$$E_s^{(V)}(t) := 2[\Psi_s^{(V)}(t) - \psi_s^{(\mathrm{true})}(t)]/(||\Psi_s^{(V)}|| + ||\psi_s^{(\mathrm{true})}||) \qquad (6.9)$$

$E_s^{(V)}(t)$ provides a relative error measurement of how well the true kinetics solution is approximated by the variational solution $\Psi_s^{(V)}(t)$ with minimized kinetic energy function H_{K}. This variational minimum-H_{K} solution is made unique by including the initial conditions $\psi_s^{(\mathrm{true})}(t_0)$ as virtual experimental data points in H_{X}, i.e., by imposing the initial conditions as a constraint in the minimization of H_{K}. In this manner, I am actually using the super-ensemble MC annealing procedure as a variational, approximate ODE solver.

Figures 6.1 and 6.2 show the results of this variational ODE solving approach. Also, in Table 6.1, I summarize the results for the corresponding values of the energy functions H_{K} and H_{X} and of the overall relative errors $||E_s^{(R)}||$ and $||E_s^{(V)}||$ before and after the MC annealing procedure. Clearly, from Figures 6.1 and 6.2 the true kinetics solution $\psi_s^{(\mathrm{true})}(t)$ is well approximated by the variational solution $\Psi_s^{(V)}(t)$, to within a relative error $||E_s^{(V)}||$ of



(a) Representation and variational errors of species E

(b) Representation and variational errors of species ES_2

(c) Representation and variational errors of species P

(d) Representation and variational errors of species S

Figure 6.2: The black lines present the representation error $E_s^{(R)}$ before MC annealing, reduced by a factor of 20 for display, while the red ones represent the variational error $E_s^{(V)}$ after MC annealing, starting from the MC initialization in Equation 6.1.

Before Annealing	After Annealing
$\vert\vert E_1^{(R)}\vert\vert = 0.014857$	$\vert\vert E_1^{(V)}\vert\vert = 0.0225404$
$\vert\vert E_2^{(R)}\vert\vert = 0.00886193$	$\vert\vert E_2^{(V)}\vert\vert = 0.0188058$
$\vert\vert E_3^{(R)}\vert\vert = 0.00187586$	$\vert\vert E_3^{(V)}\vert\vert = 0.0201145$
$\vert\vert E_4^{(R)}\vert\vert = 0.000965275$	$\vert\vert E_4^{(V)}\vert\vert = 0.0031292$
$H_{\mathrm{K}} = 0.0314609$	$H_{\mathrm{K}} = 1.86099 \times 10^-7$
$H_{\mathrm{X}} = 0$	$H_{\mathrm{X}} = 7.81489 \times 10^-7$

Table 6.1: The results of the corresponding values of the energy functions H_{K} and H_{X} and of the overall relative errors $||E_s^{(R)}||$ and $||E_s^{(V)}||$ for species $s = 1, 2, 3, 4$ corrsponding to E, ES_2, P, S respectively, before and after the MC annealing procedure.

better than 2.5% for $\mathcal{I}+1 = 16$ first order Lagrange basis functions. This is again sufficient for the super-ensemble simulations on typical noisy experimental data I would like to perform. I therefore have some confidence that the variational approach is capable of representing the ODE solutions, at least in principle.

One should caution, however, that the foregoing test of the variational MC approach is not a very stringent test of its practical utility as an ODE solver: by providing $y_{i,s}^{(\mathrm{init})} = \psi_s^{(\mathrm{true})}(t_i)$ as initial guess for the variational energy minimization, I have made it very easy for the MC annealing to find a "good" variational minimum. By contrast, in real parameter estimation simulation applications, I do *not* have a good initial guess, and the MC initial $\mathbf{Y}^{(\mathrm{init})}$ would be typical chosen more or less randomly. As a more realistic test of the variational MC approach, I should therefore start the MC annealing procedure with completely randomly chosen $\mathbf{Y}^{(\mathrm{init})}$ and then check whether the MC annealing still generates a "good" variational solution. This will be done in Section 6.2.

Likewise, in more conventional ODE solution applications, only the initial conditions $\psi_s(t_0)$ are given. To test the performance of the super-ensemble algorithm as a variational ODE solver, I have also performed a simulation in such a conventional ODE solution setting where the initial concentrations $X_s = [s](t_0)$ and rate coefficients θ_r are given and a unique

ODE solution for such a given parameter vector $\mathbf{\Gamma} = (\mathbf{\Theta}, \mathbf{X})$ is sought. This $\mathbf{\Gamma}$ is used as time-(i-)independent MC initial, i.e.,

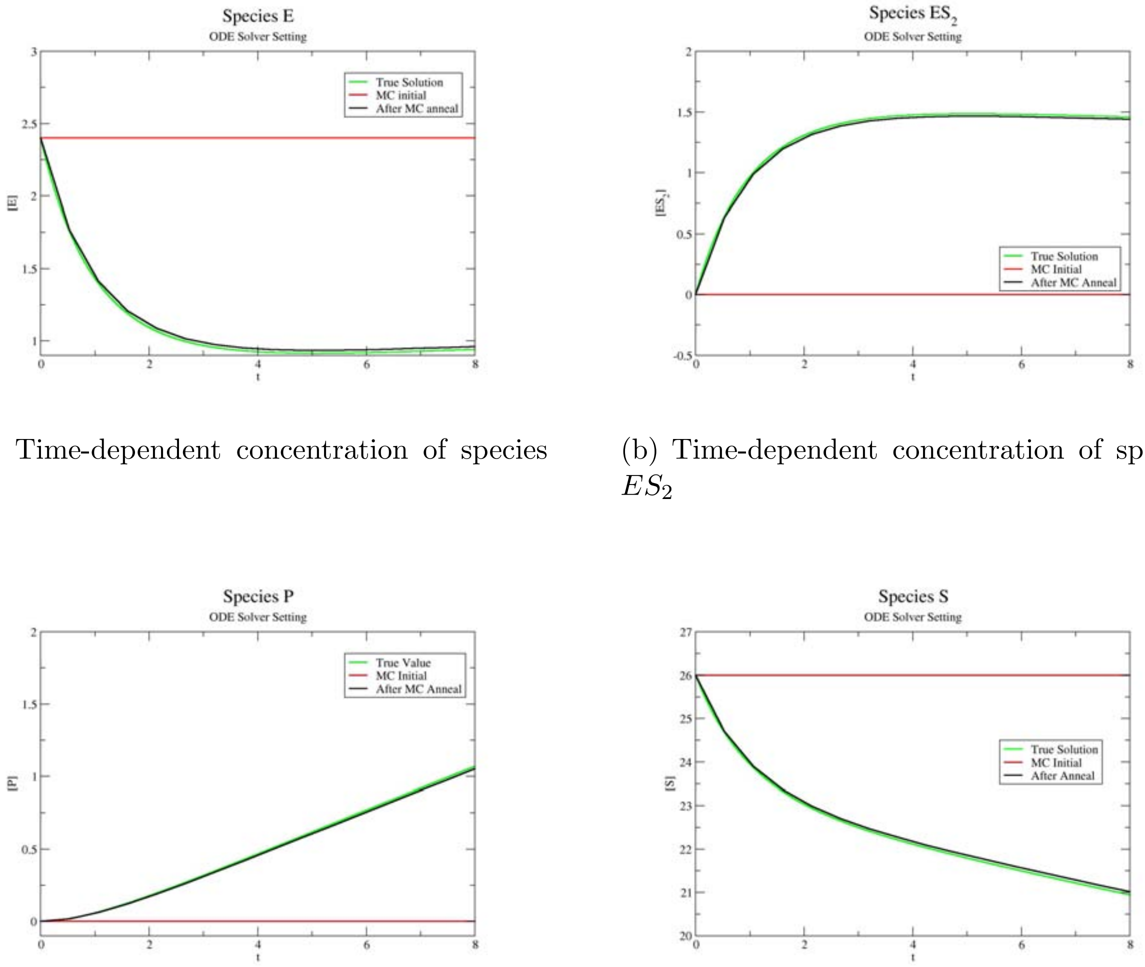
$$y_{i,s}^{(\mathrm{init})} = \begin{cases} X_s & \text{for } s = 1, \ldots, \mathcal{S} \\ \theta_{s-\mathcal{S}} & \text{for s} = \mathcal{S} + r = \mathcal{S} + 1, \ldots, \mathcal{S} + \mathcal{R} \end{cases} \tag{6.10}$$

for all $i = 0, \ldots, \mathcal{I}$. For the test below I have used again $\mathbf{\Gamma}^{(\mathbf{true})}$ from Table 5.1(a).

The standard MC annealing protocol with standard parameters described in Section 5.2.2 is performed once for $\mathcal{I} + 1 = 16$ time grid with MC repetition $\mathcal{J} = 1$. Figures 6.3 and 6.4 show the results for the $\mathbf{Y}^{(\mathrm{finl})}$ reached at the end of this MC annealing. Figures 6.3 are the species concentration $\Psi_s(t, \mathbf{Y}^{(\mathrm{finl})})$ and Figures 6.4 are the corresponding relative error, $E_s^{(\mathrm{R})}$ and $E_s^{(\mathrm{V})}$, before and after the annealing, respectively. Clearly, the variational annealing results starting from the "ODE solver" initial (Equation 6.10) is just as accurate as the ones shown in Figures 6.1 and 6.2 generated from the true kinetics solution being used as the MC initial. This result suggests that super-ensemble variational MC approach can be utilized as a parallelizable ODE solver in a conventional ODE solution setting. However, it remains to be explored, especially for larger networks, whether this approach is competitive with conventional serial ODE solver algorithms, such as the Runge-Kutta or backward differentiation approaches [5].

6.2 Super-Ensemble Monte Carlo Results

In Figures 6.5, through 6.16, I show the results for two super-ensemble simulations, using the MC protocol described in Section 3.2, with random MC initial $\mathbf{Y}^{(\mathrm{init})}$, Equation 5.7; for two choices of $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$, $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = \beta_{\mathrm{K}}^{(\mathrm{finl})} = 3.98 \times 10^8$ and $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^4$; $\mathcal{J} = 10$ MC repetitions; a first order Lagrange basis with $\mathcal{I}+1 = 16$ interpolation grid points t_i; and the zero-noise data set 0 from Table 5.2 with $\sigma_{n,s}$ set to $1\% \times Z_{ref}$ in H_{X}. Figures 6.5 show the MC protocols and Figure 6.6 the corresponding MC trajectories of the energies H_{X} and H_{K} as a function of MC sweep number for $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^8$ and 3.98×10^4. H_{X} and H_{K} have been averaged over the $\mathcal{J} = 10$ MC repetitions; the respective error bars are the standard deviations calculated from



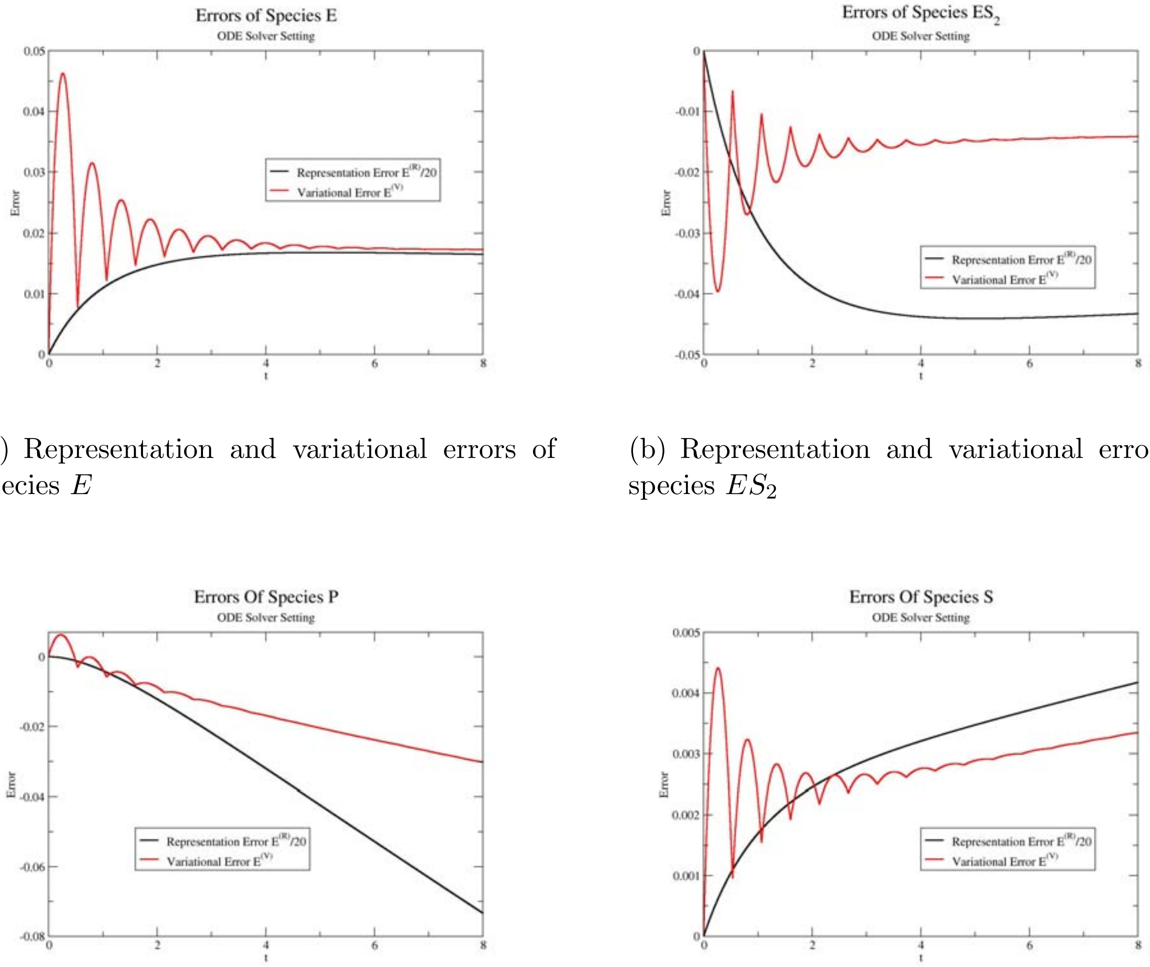
(a) Time-dependent concentration of species E

(b) Time-dependent concentration of species ES_2

(c) Time-dependent concentration of species P

(d) Time-dependent concentration of species S

Figure 6.3: The green lines represent the true kinetics solution $\psi_s^{(\text{true})}$; the black lines represent MC initial $\Psi_s^{(R)}(t, \mathbf{Y})$; and the red lines represent variational solutions after MC annealing phase $\Psi_s^{(V)}(t, \mathbf{Y})$.



(a) Representation and variational errors of species E

(b) Representation and variational errors of species ES_2

(c) Representation and variational errors of species P

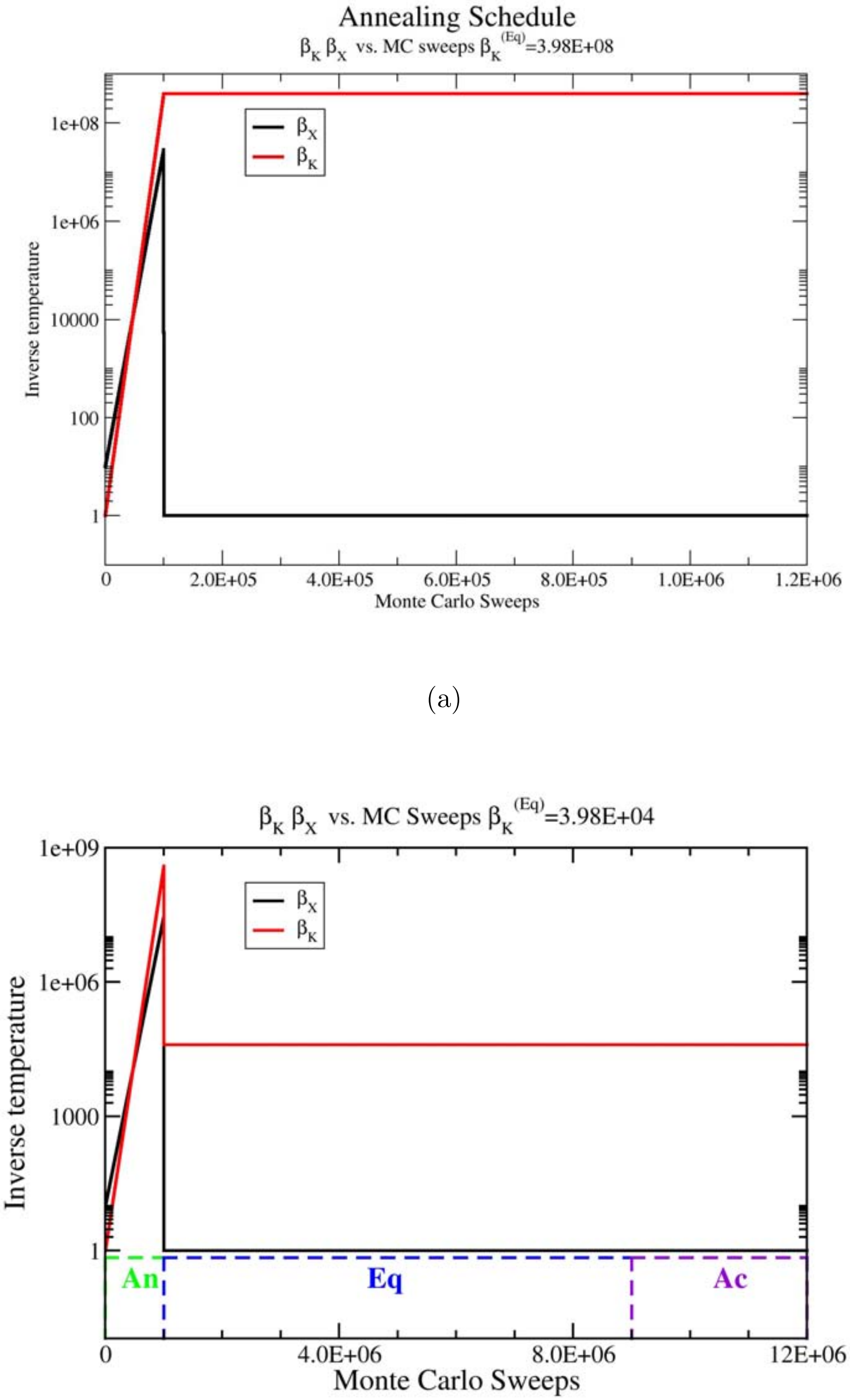
(d) Representation and variational errors of species S

Figure 6.4: The black lines present the representation the y-scale transformation of error $E_s^{(R)}$, i.e. $E_s^{(R)}$ shown in these figures is 20 times smaller, while the red ones represent the variational error $E_s^{(V)}$.

this sample of repetitions separately at fixed MC sweep number(i.e., without averaging over the MC sweep number). As expected, both H_{X} and H_{K} start at very high values, since $\mathbf{Y}^{(\mathrm{init})}$ is chosen completely randomly according to Equation 5.7 in each MC repetition. During the annealing phase, (lasting for the first 10^6 MC sweeps,) H_{X} and H_{K} both drop very rapidly. And during the equilibration phase for $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^8$, (lasting for the next 8×10^6 MC sweeps,) H_{K} continues to decrease while H_{X} increases. This is due to the reduction of β_{X} from $\beta_{\mathrm{X}}^{(\mathrm{finl})} = 3.98 \times 10^8$ to $\beta_{\mathrm{X}}^{(\mathrm{Eq})} = 1.0$ which softens the constraint by the experimental data in H_{X} and thereby allows more variational freedom to reduce H_{K}. When $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ is reduced to 3.98×10^4, H_{K} also rises in the transition from annealing to equilibration, since the ODE solution constraint imposed by H_{K} is now enforced less strongly during equilibration. By the end of the equilibration phase, both H_{K} and H_{X} appear to have reached a stable equilibrium where accumulation can commence, for both $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$.

The ensemble average $< \Psi_s(t;.) >_{MC}$ for the time-dependent real species concentrations for $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^8$ are shown in Figures 6.7 and 6.8. They agree with the experimental data to within their MC standard deviation. Since these experimental data are a noiseless sample of the true kinetics solution at a few random times $\bar{t}_n$, the MC averages $< \Psi_s(t;.) >_{MC}$ are also in reasonably good agreement with the true kinetics solution $\psi_s^{(\mathrm{true})}(t)$.

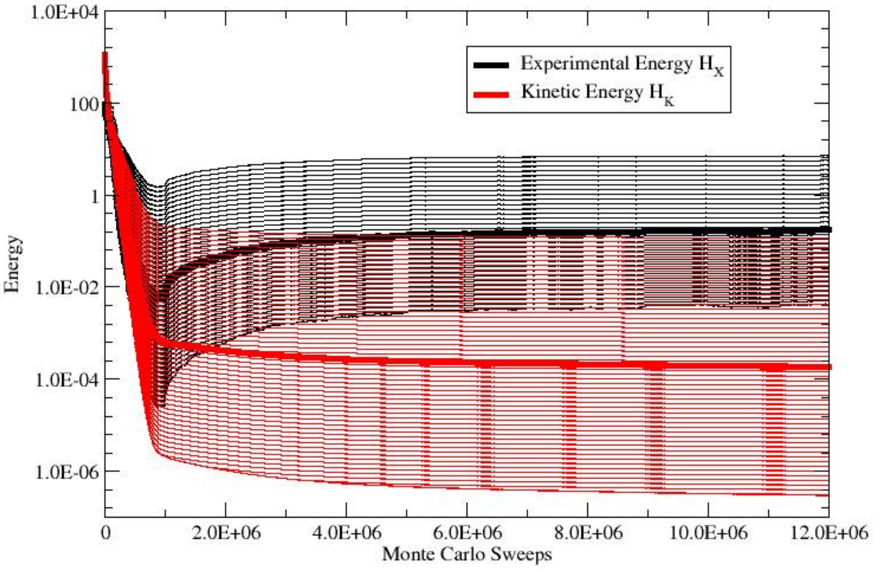
The corresponding plots of Ψ_s vs. t for the fictitious species $s = \mathcal{S} + r$ representing the reaction rate coefficients θ_r are shown in Figure 6.9. These fictitious species solutions should actually be time-independent according to Equation 4.5. However, in the simulation, they still show significant time-dependence. Furthermore, the results for the fictitious species $s = \mathcal{S} + r$ for reactions $r = 1$ and $r = 3$ are in noticeable disagreement with the true values $\theta_r^{(\mathrm{true})}$ also shown in Figure 6.9. The lack of time independence suggests that the simulation has not yet fully equilibrated, even after 8×10^6 equilibration sweeps. However, a lack of equilibration alone cannot completely account for the systematic deviation of $< \Psi_{\mathcal{S}+r}(t;.) >_{MC}$ from the true values of θ_r. If only incomplete equilibration were to blame for this discrepancy, one should also see similar discrepancies between the ensemble averages $< \Psi_s(t;.) >_{MC}$ and



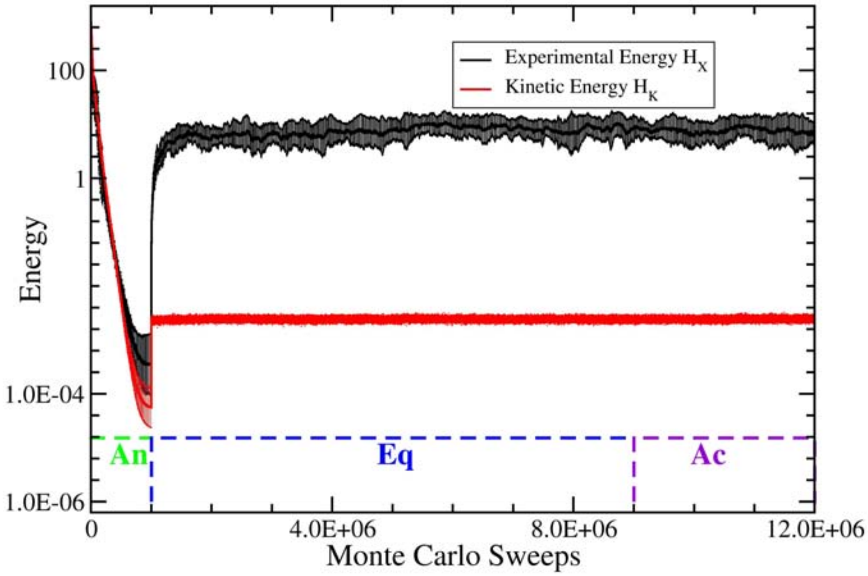
(a)

(b)

Figure 6.5: MC annealing, equilibration and accumulation schedule for $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^8$ in (a) and 3.98×10^4 in (b).

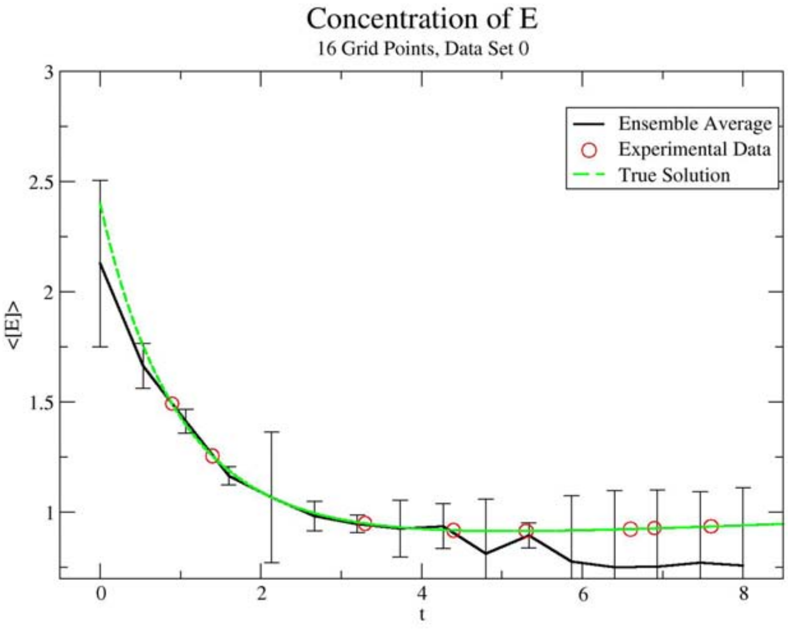


(a) $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^8$

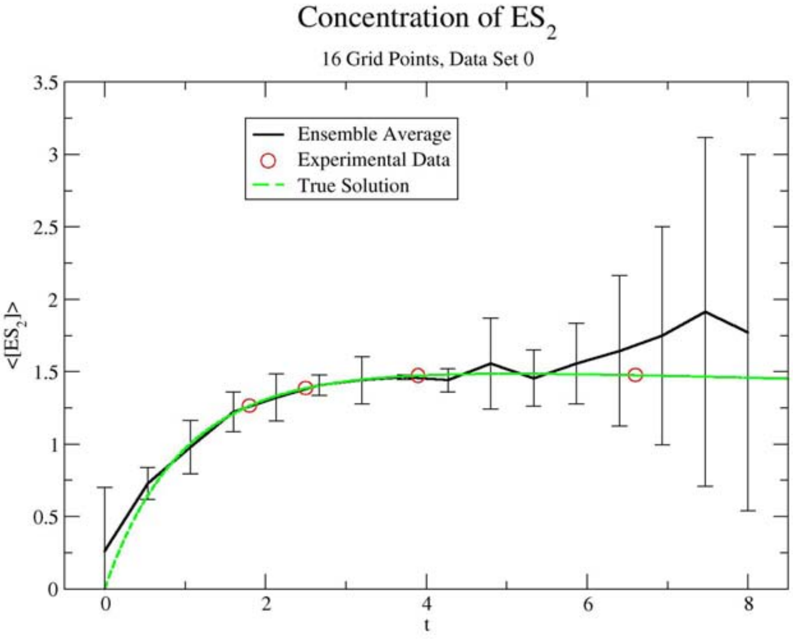


(b) $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^4$

Figure 6.6: Energy functions H_{X} and H_{K} v.s. Monte Carlo sweep number for zero-noise Data Set 0, $\omega_\sigma = 0\%$. Here $\mathcal{I} + 1 = 16$ and $\sigma_{n,s} = 1\% \times Z_{ref}$ in H_{X}, $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^8$ in (a) or 3.98×10^4 in (b), using the MC protocols shown in Figures 6.5(a) and 6.5(b), respectively, with MC annealing parameters $\beta_{\mathrm{K}}^{(\mathrm{init})} = 1.0$, $\beta_{\mathrm{K}}^{(\mathrm{finl})} = 3.98 \times 10^8$, $\beta_{\mathrm{X}}^{(\mathrm{init})} = 1.352$, and $\beta_{\mathrm{X}}^{(\mathrm{finl})} = 3.95 \times 10^6$, or given in Equations 5.15 to 5.17.

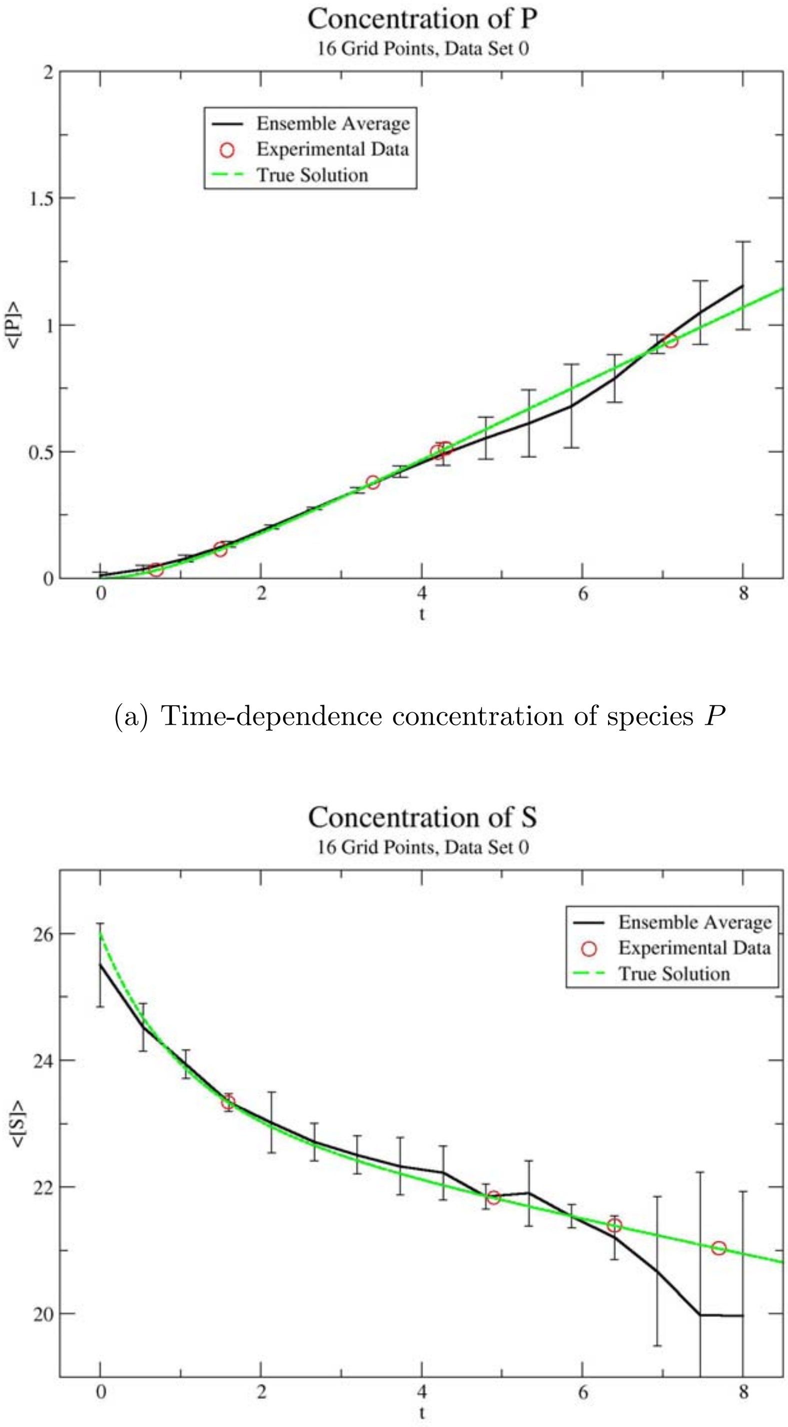


(a) Time-dependence concentration of species E



(b) Time-dependence concentration of species ES_2

Figure 6.7: Black lines are ensemble averages for species E and ES_2 along with ESD for the MC run shown in Figures 6.5(a) and 6.6(a). Red circles present experimental data set 0, $\omega_\sigma = 0\%$. Green dash lines are true solutions using standard numerical solution to ODE with "true" initial concentrations and reaction coefficients listed in Table 5.1(a).



(a) Time-dependence concentration of species P

(b) Time-dependence concentration of species S

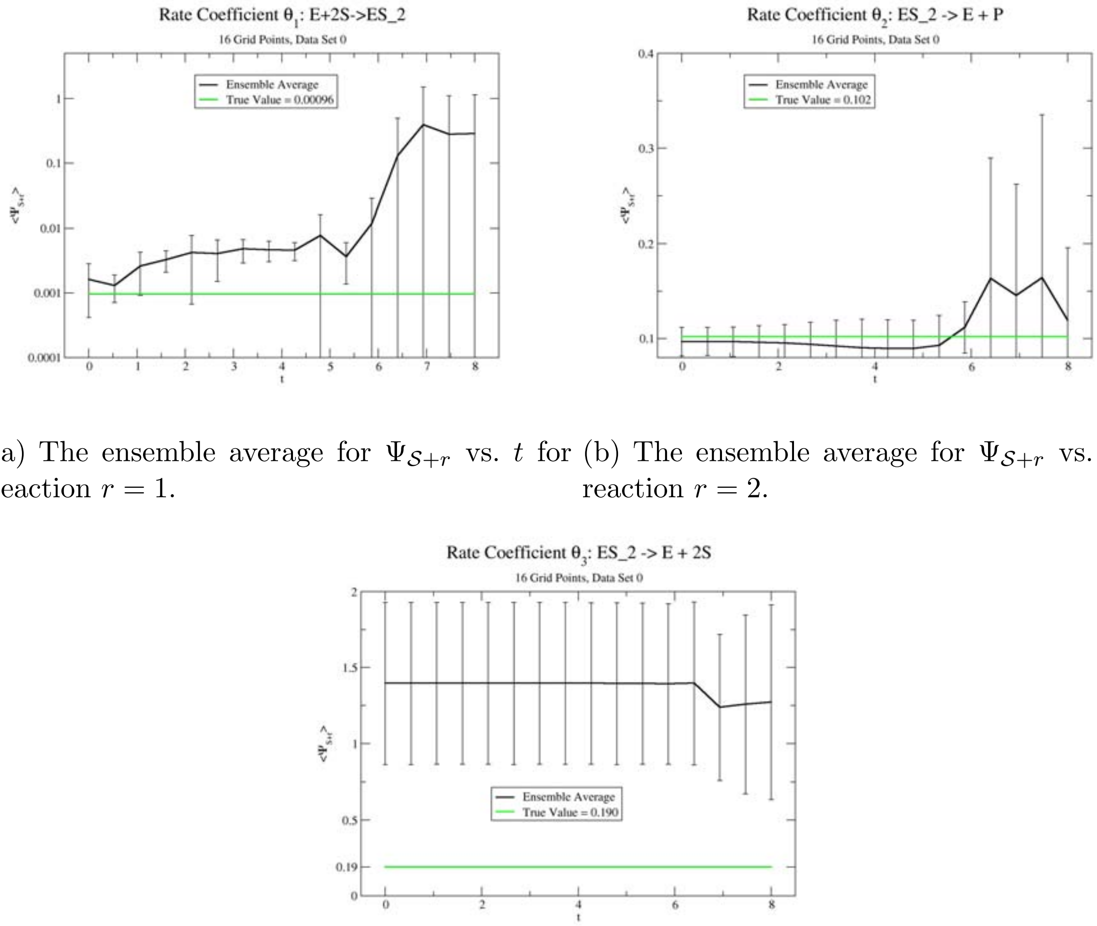
Figure 6.8: Black lines are ensemble averages for species S and P along with ESD for the MC run shown in Figures 6.5(a) and 6.6(a). Red circles present experimental data set 0, $\omega_\sigma = 0\%$. Green dash lines are true solutions using standard numerical solution to ODE with "true" initial concentrations and reaction coefficients listed in Table 5.1(a).

the true kinetics solutions. Evidently, as shown in Figures 6.7 and 6.8, there are no such large discrepancies for the real species. This suggests that the discrepancies between the ensemble and the true values for the rate coefficients might be caused by systematic errors, for example due to the FE basis representation or due to entropy effects. The ensemble MC process could for example try to compensate for systematic FE representation errors by shifting the θ_r values so as to produce variational $\Psi_s(t; \mathbf{Y})$ solutions which best minimize H_{X}. In this scenario, the ensemble averages $< \Psi_s(t; .) >_{MC}$ would still match the experimental data, albeit at the expense of the converging to the "wrong" θ_r values.

Another potential source of systematic error lies in possible entropy effects arising from the finite H_{X} simulation "temperature", $1/\beta_{\mathrm{X}} > 0$, i.e. from the fact that $\beta_{\mathrm{X}} < \infty$. For $1/\beta_{\mathrm{X}} \rightarrow 0$, a perfectly equilibrated MC sample should include only the "ground state" configuration of H_{X}, where $< \Psi_s(t; .) >_{MC}$ is close to the "best" possible variational approximation to the true kinetics solution, since the zero-noise experimental data used in H_{X} do represent the "true" kinetics solution. [Figures 6.3 illustrate a $\Psi_s(t; \mathbf{Y})$ very close to this ground state.] However, for finite H_{X} simulation temperature, $1/\beta_{\mathrm{X}} > 0$, entropy effects, i.e., the thermal availability of configurations $\mathbf{Y}$ that are *not* in the ground state, can systematically shift ensemble averages, for a perfectly equilibrated MC sample, away from their "true" values, in spite of the fact, that the experimental data in H_{X} do exactly represent the true kinetics solution. I will discuss below that entropy effects are most likely responsible for the deviations from the "true" values observed here.

Histograms for the various kinetics model parameter variables ($\mathbf{\Theta}$ and $\mathbf{X}$) were also collected during the MC accumulation phase, along with all MC averages. They are shown in Figure 6.10 for the initial concentrations of the real species

$$[s](t_0) = X_s = y_{i,s} \quad \text{at } i = 0 \tag{6.11}$$



(a) The ensemble average for $\Psi_{\mathcal{S}+r}$ vs. t for reaction $r = 1$.

(b) The ensemble average for $\Psi_{\mathcal{S}+r}$ vs. t for reaction $r = 2$.

(c) The ensemble average for $\Psi_{\mathcal{S}+r}$ vs. t for reaction $r = 3$.

Figure 6.9: Black lines are ensemble averages for virtual species $\Psi_{\mathcal{S}+r}(t)$ along with ESD for the MC run shown in Figures 6.5(a) and 6.6(a). Red circles present experimental data set 0, $\omega_\sigma = 0\%$. Green dash lines are true solution using standard numerical solution to ODE with "true" initial concentrations and reaction coefficients listed in Table 5.1(a).

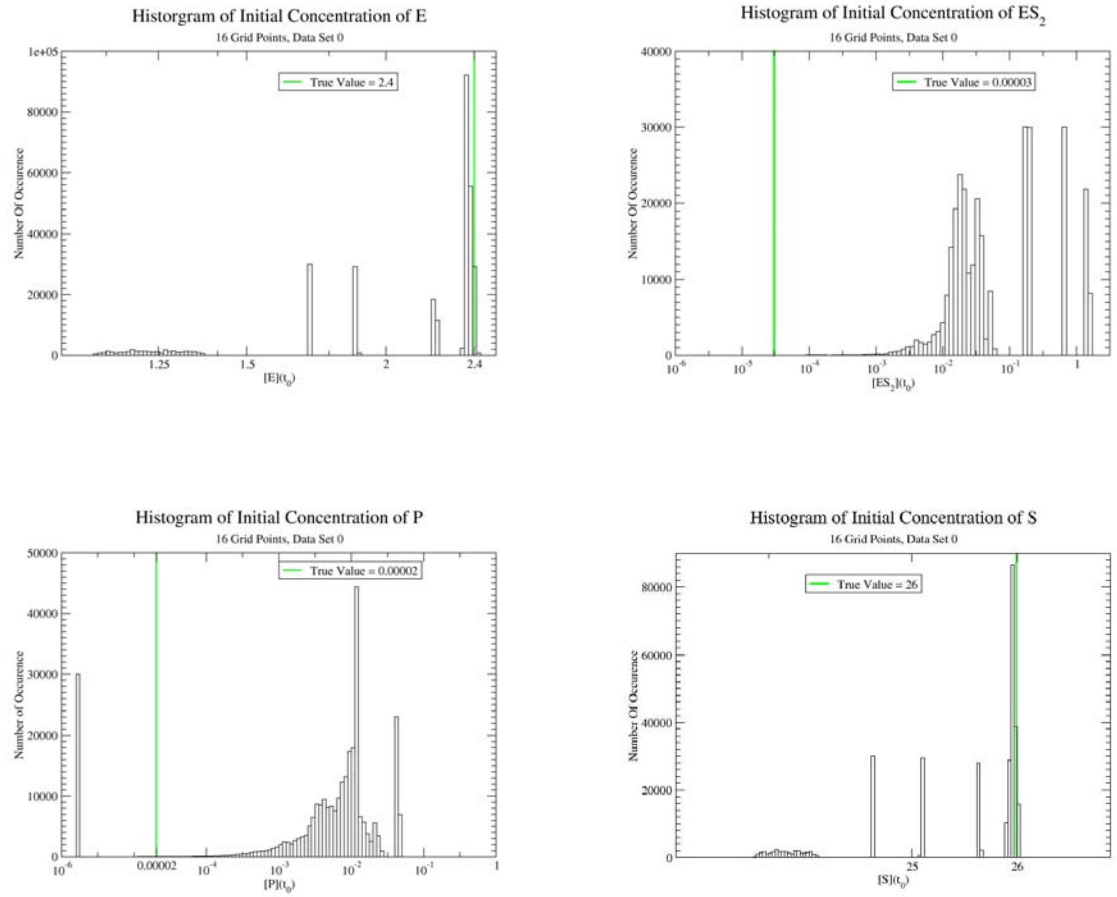


Figure 6.10: Histograms of initial concentrations of species E, ES_2, P and S for the MC run shown in Figures 6.5(a) and 6.6(a).

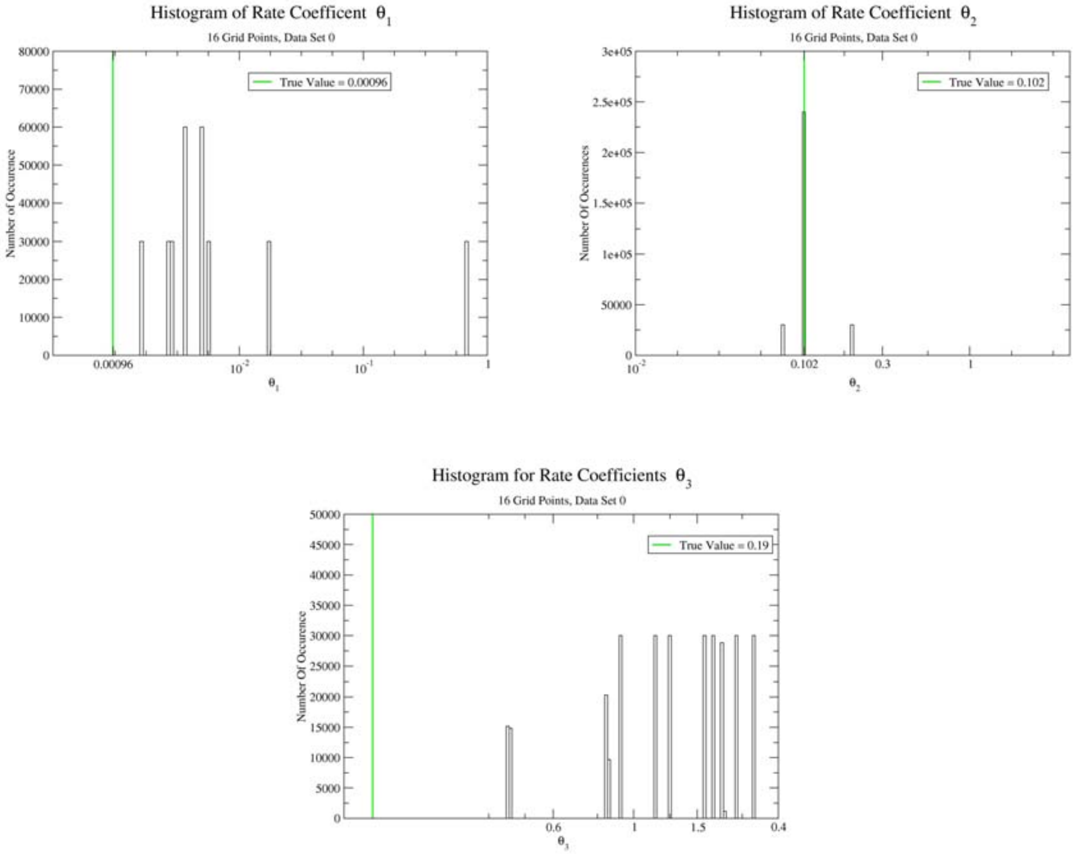


Figure 6.11: Histograms of the rate coefficients θ_r calculated from their fictitious species $\Psi_{\mathcal{S}+r}$ according to Equation 6.12 for the MC run shown in Figures 6.5(a) and 6.6(a).

and in Figure 6.11 for the rate coefficient variables θ_r, which are estimated from the time-average of their fictitious species concentration $\Psi_{\mathcal{S}+r}$, i.e., by

$$\theta_r(\mathbf{Y}) = \frac{1}{\mathcal{I}+1}\sum_{i=0}^{\mathcal{I}} \Psi_{\mathcal{S}+r}(t_i, \mathbf{Y}) = \frac{1}{\mathcal{I}+1}\sum_{i=0}^{\mathcal{I}} y_{i,\mathcal{S}+r} \tag{6.12}$$

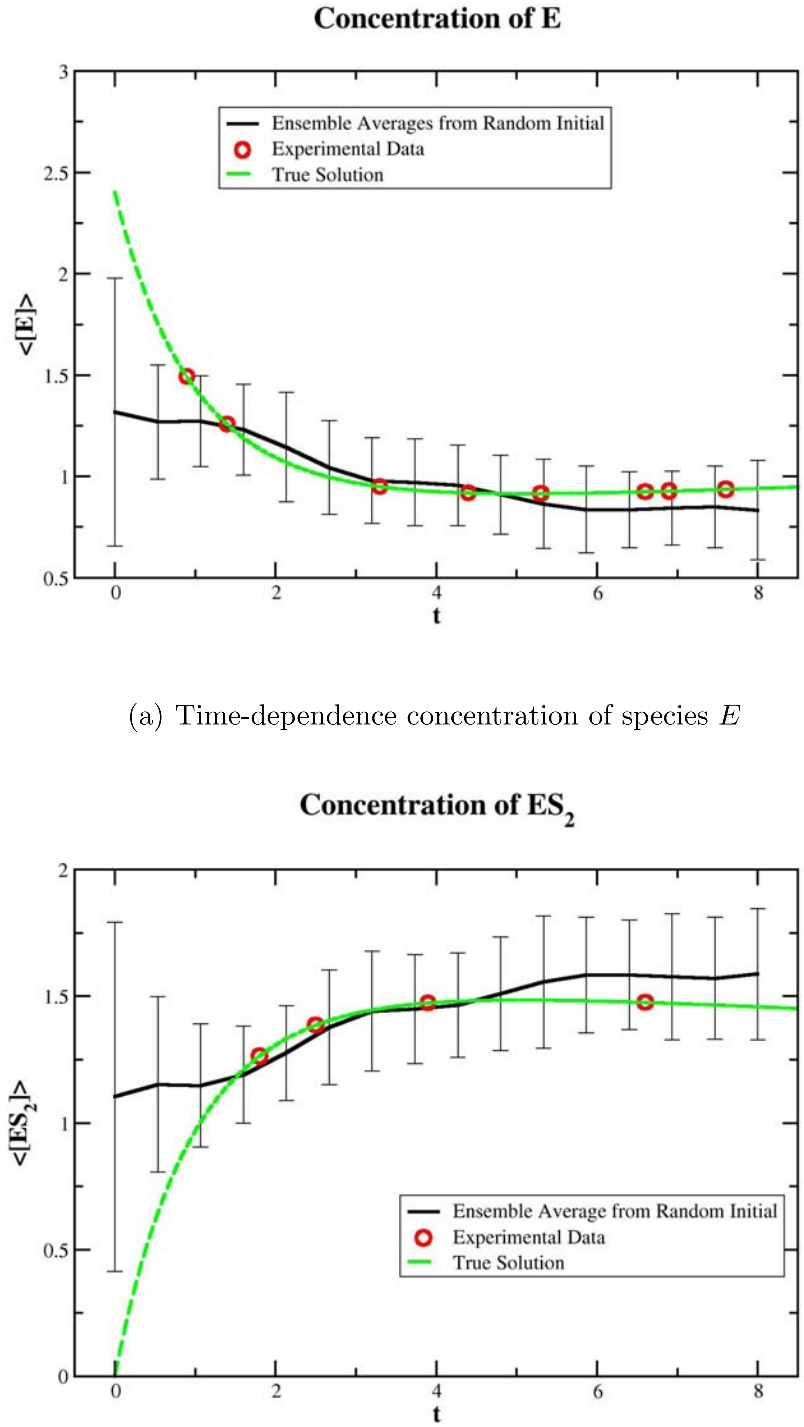
The initial concentration histograms are in reasonable agreement with the corresponding true values. One must keep in mind that species P and ES_2 have very small initial concentrations. These are not very tightly constrained by H_{X}, since even a change by a factor of 10 or more at $t = t_0$ will have very little effect on $[P](t)$ and $[ES_2](t)$ at the later times $\bar{t}_n$ that are being "probed" by H_{X}. Entropy effects discussed above can therefore easily shift the ensemble results for these small initial concentrations. The ensemble can therefore not be expected to provide an accurate reconstruction of the very small true values since the kinetic solution is very insensitive even to relatively large changes in these very small initial concentrations. Only reasonable upper limits can be obtained from the ensemble in these cases. The ensemble tells us that these values are "very" small, compared to the overall magnitude of the concentrations of P and ES_2 (which is around 2-10), but it doesn't tell us how small they are. Similarly the histograms for two of the three rate coefficients in Figure 6.11 show a very noticeable disagreement with the true values $\theta_r^{(\mathrm{true})}$ of the experimental data for $r = 1$ and $r = 3$, as already found for the corresponding fictitious species averages $< \Psi_{\mathcal{S}+r}(t; .) >_{MC}$ in Figure 6.9. I will now try to find or rule out possible causes for these discrepancies.

A possible source of incomplete equilibration in the MC simulations could be the constraints imposed on local MC moves $y_{i,s} \to y'_{i,s}$ by the "stiff" restoring forces arising form $\beta_{\mathrm{K}} H_{\mathrm{K}}$ when the equilibration value of $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ is set too high. For example, the time-dervivative term $\dot{\Psi}_s(\hat{t}_k, \mathbf{Y})$ entering into H_{K} in Equation 3.17 generates a tight coupling between the $y_{i,s}$-variables at neighboring time slices i and $i+1$. This coupling along the time axis prevents $y_{i,s}$ from being moved, unless its temporal neighbors $y_{i-1,s}$ and $y_{i+1,s}$ are moved by about the same amount, so as to keep the derivatives $\dot{\Psi}_s(\hat{t}_k, \mathbf{Y})$ in $[t_{i-1}, t_{i+1}]$ approximately constant. This coupling along the time axis is quite similar to the couplings along the chain encountered between monomer units in continuum-models of polymers. These intra-chain couplings in polymers prevent a single monomer (or smaller subunit) from being moved, unless neighboring units are moving along with it during MC updates. To overcome the equilibration problem, the super-ensemble MC simulation approach, especially in larger model networks, may well require non-local or other MC updating techniques, similar to those used in continuum-model polymer simulations. These techniques go beyond the simple local Metropolis updating scheme and I will not consider them here.

Instead, I will try to improve the equilibration behavior by simply reducing $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ and thereby softening the constraints against local MC moves. In Figures 6.6(b) and in 6.12 to 6.16, I show results obtained from an MC run with reduced $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$-value, $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^4$ using the MC protocol in Figure 6.5(b). As can be seen in Figures 6.12 and 6.13, in comparison to the corresponding $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^8$ results in Figures 6.7 and 6.8, the agreement between ensemble MC averages $< \Psi_s(t; .) >_{MC}$ for real species and true kinetics solution deteriorates somewhat when $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ is lowered. This is not a very big effect and it is not surprising, since reducing $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ weakens the ODE solution constraint imposed by H_{K} and therefore allows greater fluctuations in $\mathbf{Y}$. Consequently, the agreement between MC averages and the zero-noise experimental data points also deteriorates.

The histograms for $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^4$, in Figure 6.15 and Figures 6.16 clearly show the effect of improved equilibration, when compared to Figures 6.10 and 6.11: the sampled values of each MC variable are much more evenly dispersed across the histogram sampling interval; there is a “dense forest” of histogram bars now instead of a “few trees” in Figures 6.10 and 6.11. However, for the “bad actors” in Figure 6.7(b) and 6.8(a) and in Figure 6.16, (θ_1 and θ_3) and the corresponding $< \Psi_{\mathcal{S}+r}(t; .) >_{MC}$ in Figure 6.14, the agreement with the “true” values has not been improved by the improved MC equilibration.

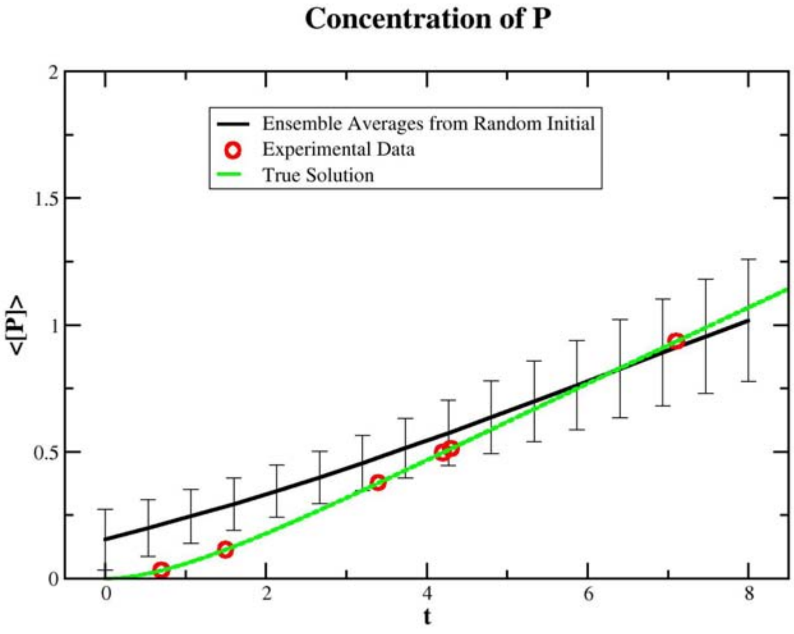
In the case of the “bad” initial concentrations, entropy effects and the poor constraints by H_{X} can easily explain the discrepancies. However, the foregoing results do of course not rule out the possibility that the discrepancies between ensemble MC results and true values for θ_1 and θ_3 could still be caused by inadequate equilibration, since the actual equilibration



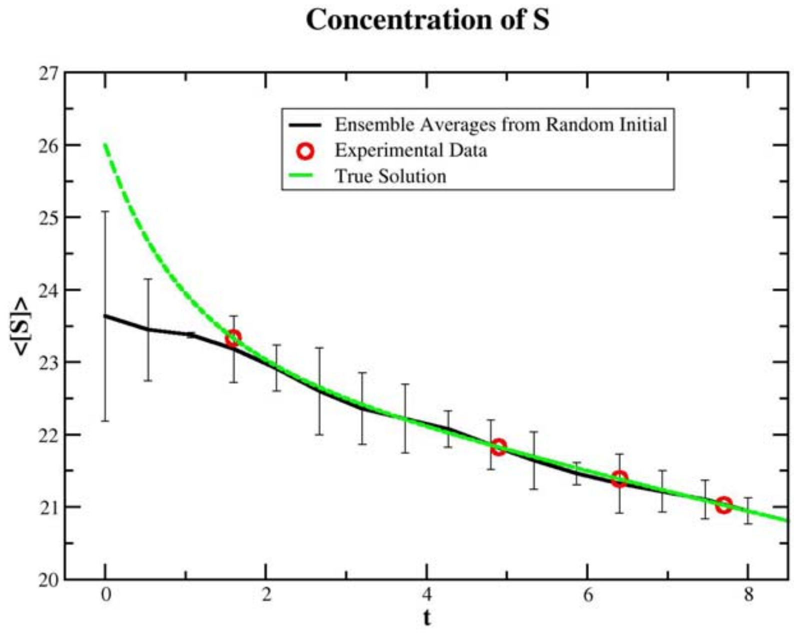
(a) Time-dependence concentration of species E

(b) Time-dependence concentration of species ES_2

Figure 6.12: Black lines are ensemble averages for species E and ES_2 along with ESD for the MC run shown in Figures 6.5(b) and 6.6(b). Red circles present experimental data set 0, $\omega_\sigma = 0\%$. Green dash lines are true solutions using standard numerical solution to ODE with "true" initial concentrations and reaction coefficients listed in Table 5.1(a).

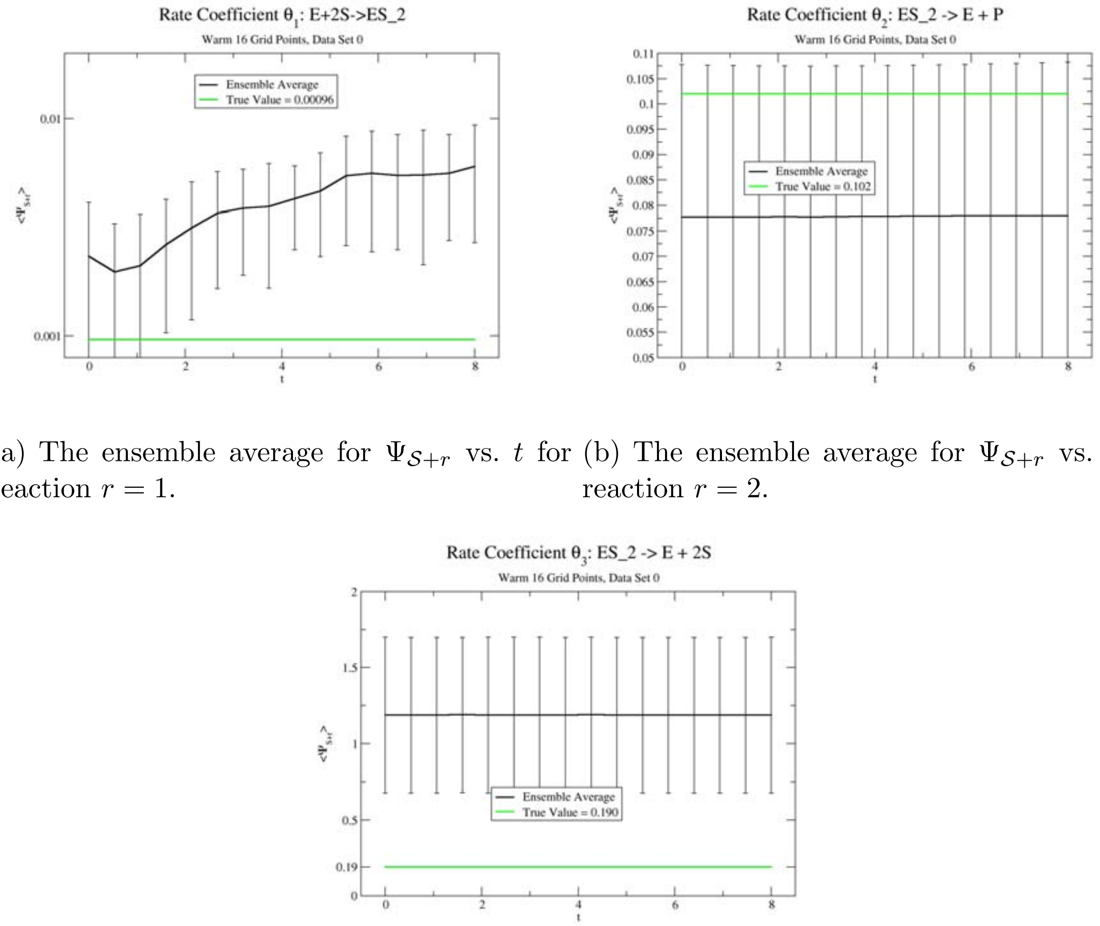


(a) Time-dependence concentration of species P



(b) Time-dependence concentration of species S

Figure 6.13: Black lines are ensemble averages for species S and P along with ESD for the MC run shown in Figures 6.5(b) and 6.6(b). Red circles present experimental data set 0, $\omega_\sigma = 0\%$. Green dash lines are true solutions using standard numerical solution to ODE with "true" initial concentrations and reaction coefficients listed in Table 5.1(a).

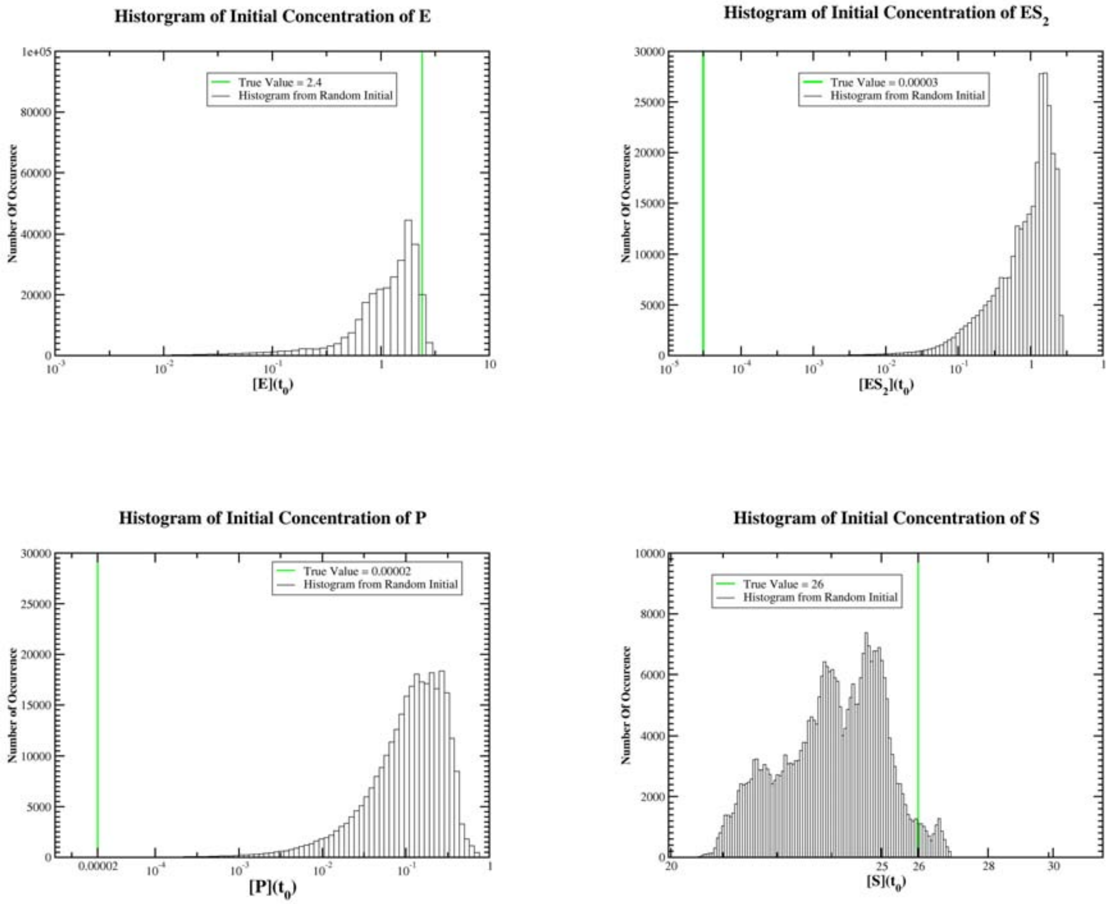


(a) The ensemble average for $\Psi_{\mathcal{S}+r}$ vs. t for reaction $r = 1$.

(b) The ensemble average for $\Psi_{\mathcal{S}+r}$ vs. t for reaction $r = 2$.

(c) The ensemble average for $\Psi_{\mathcal{S}+r}$ vs. t for reaction $r = 3$.

Figure 6.14: Black lines are ensemble averages for virtual species $\Psi_{\mathcal{S}+r}(t)$ along with ESD for the MC run shown in Figures 6.5(b) and 6.6(b). Red circles present experimental data set 0, $\omega_\sigma = 0\%$. Green dash lines are true solution using standard numerical solution to ODE with “true” initial concentrations and reaction coefficients listed in Table 5.1(a).



Historgram of Initial Concentration of E

Histogram of Initial Concentration of ES$_2$

Histogram of Initial Concentration of P

Histogram of Initial Concentration of S

Figure 6.15: Histograms of initial concentrations of species E, ES_2, P and S for the MC run shown in Figures 6.5(b) and 6.6(b).

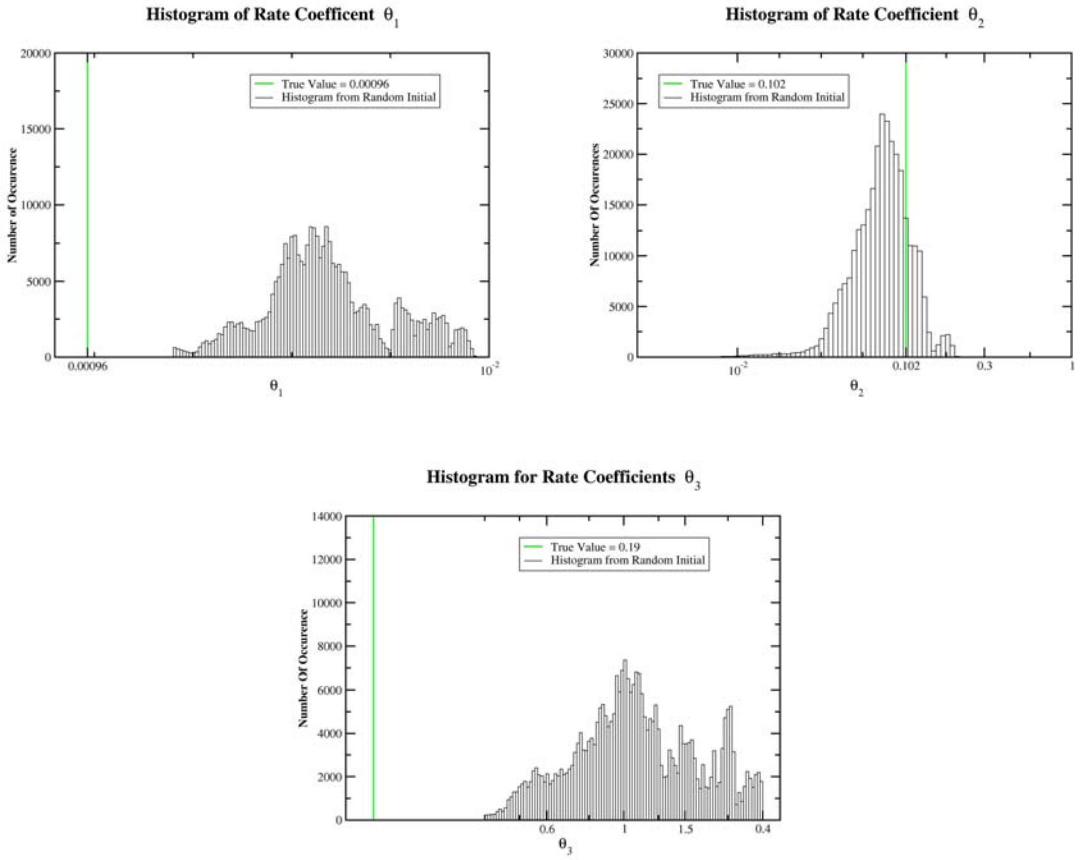


Figure 6.16: Histograms of the rate coefficients θ_r calculated from their fictitious species $\Psi_{\mathcal{S}+r}$ according to Equation 6.12 for the MC run shown in Figures 6.5(b) and 6.6(b).

times of the MC Markov chain process could, potentially, exceed the simulations by *many* order of magnitude. One should therefore also explore whether these discrepancies in θ_1 and θ_3 could be caused by systematic errors, due to the FE basis representation and due to finite $1/\beta_{\mathrm{X}}$ entropy effects, which are not related to MC equilibration.

To test for systematic errors introduced by the FE basis representation, I have performed simulations with improved FE bases, by setting $I+1=32$ and $I+1=64$ FE interpolation grid sizes, keeping all other MC parameters, and experimental input data the same as in Figures 6.7 to 6.11. The results of those simulations, not shown here in detail, are very similar to those obtained with $I+1=16$. They strongly suggest that systematic approximation errors due to the finite FE basis are negligible. That is not surprising in light of the FE basis test results discussed in Section 6.1.

To explore whether entropy effects could cause the observed systematic errors, I have performed a series of simulations starting from a different MC initialization $\mathbf{Y}^{(\mathrm{init})}$, namely the “true kinetics” initialization given in Equation 6.1 using again zero-noise experimental data. The simulation is thus started in a configuration $\mathbf{Y}$ that is at or very close to the ground state of H_{X} and in the course of the equilibration one can “watch” how the MC Markov process tries to *escape* from the H_{X}-ground state energy minimum when equilibrating at finite temperature $1/\beta_{\mathrm{X}}=1.0$. This is in contrast to the random MC initialization runs, discussed above, where $\mathbf{Y}$ is started far away from the H_{X}-ground state and during equilibration the MC Markov process tries to *find* the minimum. The results from these simulations, again not shown here in detail, suggest that entropic effects do indeed contribute substantially to the discrepancies from the true values observed in θ_1 and θ_3.

The main finding from these “true-kinetics”-initializaed MC runs is that it really does not matter whether the MC is initialized *close* to the ground state of H_{X} or *far away* from it. For either initialization, the MC Markov process will eventually “escape” from or, respectively, fail to find the near-ground state region in $\mathbf{Y}$-space during equilibratioin when the simulation temperature $1/\beta_{\mathrm{X}}$ has been raised to 1.0 and $\beta_{\mathrm{K}}^{(\mathrm{Eq})}$ has been reduced to 3.98×10^{4} to allow for improved equilibration. After equilibration, the MC samples and averages for θ_1 and θ_3, as well as all other quantities, are quite similar for the two different initializations.

As it turns out, θ_1 and θ_3 are poorly constrained by H_{X}, even though θ_2 is well constrained. This can be understood in terms of a chemical quasi-equilibrium [17, 18] that establishes itself between reaction $r = 1 : E + 2S \rightarrow ES_2$ and its time-reversed backward reaction $r = 3 : ES_2 \rightarrow E + 2S$. In this quasi-equlibrium state, the rates of $r = 1$ and $r = 3$ are almost balanced so that (See Equations 2.1 and 2.3), approximately:

$$\theta_1[E][S]^2 \simeq \theta_3[ES_2] \tag{6.13}$$

Hence, the 3 species involved approximately obey the law of mass action [17, 18]

$$\frac{[E][S]^2}{[ES_2]} \simeq \frac{\theta_3}{\theta_1} \tag{6.14}$$

even though the system is not really in equilibrium since $[E]$, $[S]$ and $[ES_2]$ are actually changing with time t. In this quasi-equilibrium state the individual values of θ_1 and θ_3 are poorly constrained by the experimental concentration data for $[E]$, $[ES_2]$ and $[S]$: according to Equation 6.14, only the ratio θ_3/θ_1 is well constrained by the concentration data. The MC updating process can scale the values of θ_1 and θ_3 up or down by an arbitrary common scale factor without substantially changing the agreement with the experimental data, and hence, without substantial change in energy H_{X}. However for larger (scaled-up) values of θ_1 and θ_3, the available "phase space" volume is also scaled up. Consequently, at finite temperature $1/\beta_{\mathrm{X}} > 0$, entropy favors values of θ_1 and θ_3 that are both too large, compared to the true values.

To test this quasi-equilibration scenario, I have calculated the MC averages and histograms of the right-hand side and the left hand side of Equation 6.14. To within error bars they agree with each other and the true value, that is

$$< \frac{\theta_3}{\theta_1} > \simeq \frac{\theta_3^{(\mathrm{true})}}{\theta_1^{(\mathrm{true})}} \tag{6.15}$$

as expected.

I have also performed super-ensemble simulations for the "noisy" virtual data sets, Data Set 1 and Data Set 2 listed in Table 5.2, using the same Monte Carlo protocol and grid size $\mathcal{I}+1 = 16$ as for the simulations on the zero-noise data set show in Figures 6.7 to 6.11, with $\beta_{\mathrm{K}}^{(\mathrm{Eq})} = 3.98 \times 10^8$. The results are consistent with those for the zero-noise data discussed above. The primary effect of the noise is to broaden the ensemble distribution of most quantities $A(\mathbf{Y})$, including, e.g., the species concentrations $\Psi_s(t;\mathbf{Y})$. This indicates that experimental noise tends to increase the overall entropy of the ensemble distribution. As a consequence, also the ensemble standard deviations are increased, as are the deviations between ensemble average and true results for the time-dependent species concentrations. Not surprisingly the entropy-induced discrepancies between ensemble predictions and true values, e.g., for θ_1 and θ_3 persist. Other model parameters are still reasonably well reconstructed from both data sets. If the data noise level is increased to $\sigma_{n,s} = 5\% \times Z_{ref}$, i.e., substantially higher levels than for Data Set 1 and 2 in Table 5.2, one loses the ability to reconstruct any model parameter values from the noisy data. Apparently, at that noise level, there is essentially no information content left in the data.

6.3 Parallelization Speedup

To test the efficiency of the parallelization method I have described in Section 4.2, I have also performed a series of simulation runs on different interpolation grid sizes $\mathcal{I}+1 = 16$, 32, 64, with different processor numbers $\mathcal{P} = 1$, 2, 4, 8. The results are shown in Figure 6.17 represented in terms of the so-called parallelization speed-up, defined as

$$\text{speed-up} = T(1)/T(\mathcal{P}) \tag{6.16}$$

Here $T(\mathcal{P})$ is the amount of CPU time consumed by a single processor during a simulation run with $\mathcal{P}$ processors working in parallel.

Theoretically, I would expect $T(\mathcal{P})$ to scale approximately with processor number $\mathcal{P}$ and and time grid size $(\mathcal{I}+1)$ as follows:

$$T(\mathcal{P}) = T_0 + T_C\frac{\mathcal{I}+1}{\mathcal{P}} + T_{MS}\mathrm{log}_2(\mathcal{P}) + T_{SS} \tag{6.17}$$

where T_{MS}, T_{SS}, T_C and T_0 are constant independent of $\mathcal{P}$ and $\mathcal{I}+1$. The T_C term is the intra-processor computation time. This should be proportional to the time sub-domain size $\Delta i = (\mathcal{I}+1)/\mathcal{P}$ assigned to each processor. The T_{MS}-term represents the communication overhead for broadcasting data, such as the MC sweep number l, from the master to all slave processors. This is also required to properly synchronize all slaves at the end of each MC sweep. Efficient broadcasting algorithms typically employed in modern parallel machines scale as $\log_2(\mathcal{P})$. The T_{SS} term represents the slave-to-slave communication time for sending the $y_{i,s}$-values from the most recently updated boundary layers $B^{\pm}(p)$ to the neighbor processor across the boundary. This boundary layer communication is done in parallel, i.e., with all processors p sending their respective $y_{i,s}$-values to their neighbors $p \pm 1$ simultaneously. The T_{SS} term is therefore independent of $\mathcal{P}$. It is also independent of $(\mathcal{I}+1)$ since the boundary layer size depends only on the network size $(\mathcal{S}+\mathcal{R})$ but not no the time grid size $(\mathcal{I}+1)$. The T_0 term presents $\mathcal{P}$ independent contribution for miscellaneous initial start-up overhead of the simulation and the non-parallelizable part of algorithm [1].

From Equation 6.17.

$$\frac{T(1)}{T(\mathcal{P})} = \mathcal{P}\frac{1 + [(T_{SS}+T_0)/T_C](\mathcal{I}+1)^{-1}}{1 + [T_{MS}\mathrm{log}_2\mathcal{P} + T_{SS} + T_0]/T_C]\mathcal{P}(\mathcal{I}+1)^{-1}} \tag{6.18}$$

So a linear speed-up, with

$$\frac{T(1)}{T(\mathcal{P})} \simeq \mathcal{P} \tag{6.19}$$

is expected for sufficiently large time sub-domain size

$$\Delta i = \frac{(\mathcal{I}+1)}{\mathcal{P}} \gg \frac{T_{MS}\mathrm{log}_2(\mathcal{P}) + T_{SS} + T_0}{T_C} =: \Delta_{\mathrm{lin}} \tag{6.20}$$

When $\mathcal{P}$ reaches values comparable to

$$\mathcal{P}_{\mathrm{lin}} := \frac{\mathcal{I}+1}{\Delta i_{\mathrm{lin}}} \tag{6.21}$$

the speed-up will become sub-linear. According to Equation 6.18, the speed-up would reach a hypothetical maximum for some $\mathcal{P}_{max}$ several times larger than $\mathcal{P}_{\text{lin}}$ and it would decrease as $1/\log_2(\mathcal{P})$ when $\mathcal{P} \gg \mathcal{P}_{max}$. However, most likely, that $\mathcal{P}_{max}$ can never be reached, since the sub-domain size Δi cannot be less than twice the range i_Φ of the FE basis functions, according to Equation 4.30. Therefore $\mathcal{P}$ is limited to values

$$\mathcal{P} \leq \mathcal{P}_\Phi := \frac{\mathcal{I}+1}{2i_\Phi} \tag{6.22}$$

Any available processor in excess of $\mathcal{P}_\Phi$ cannot be employed in the time-domain decomposition algorithm and no further speed-up can be achieved by adding more processors beyond $\mathcal{P}_\Phi$. So for $\mathcal{P} = \mathcal{P}_\Phi$, the speed-up reaches its actually achievable maximum value

$$\frac{T(1)}{T(\mathcal{P}_\Phi)} = \frac{\mathcal{I}+1}{2i_\Phi} \frac{1 + [(T_{SS} + T_0)/T_C](\mathcal{I}+1)^{-1}}{1 + [T_{MS}\log_2(\frac{\mathcal{I}+1}{2i_\Phi}) + T_{SS} + T_0]/T_C](2i_\Phi)^{-1}} \tag{6.23}$$

Results shown in Figure 6.17 are consistent with the foregoing scaling arguments: for some $\mathcal{P}$-range, the speed-up is approximately linear and that range of linearity increases with increasing time domain size $\mathcal{I}+1$. For the largest $\mathcal{P} = 8$ and the smaller domain sizes ($\mathcal{I}+1 = 16, 32$) sub-linear speed-up is observed.

For the smaller processor numbers, $\mathcal{P} = 2, 4$, the speed-up in Figure 6.17 is actually slightly *super*-linear. I do not currently understand the cause of this. However, a possible explanation could be the incertainties of the CPU-time measurements. Since these test runs were performed on the UGA IBM Pcluster system, the processors may or may not have been shared with other users during execution which could affect the CPU time in an irreproducible manner during different runs with different $\mathcal{P}$ numbers.

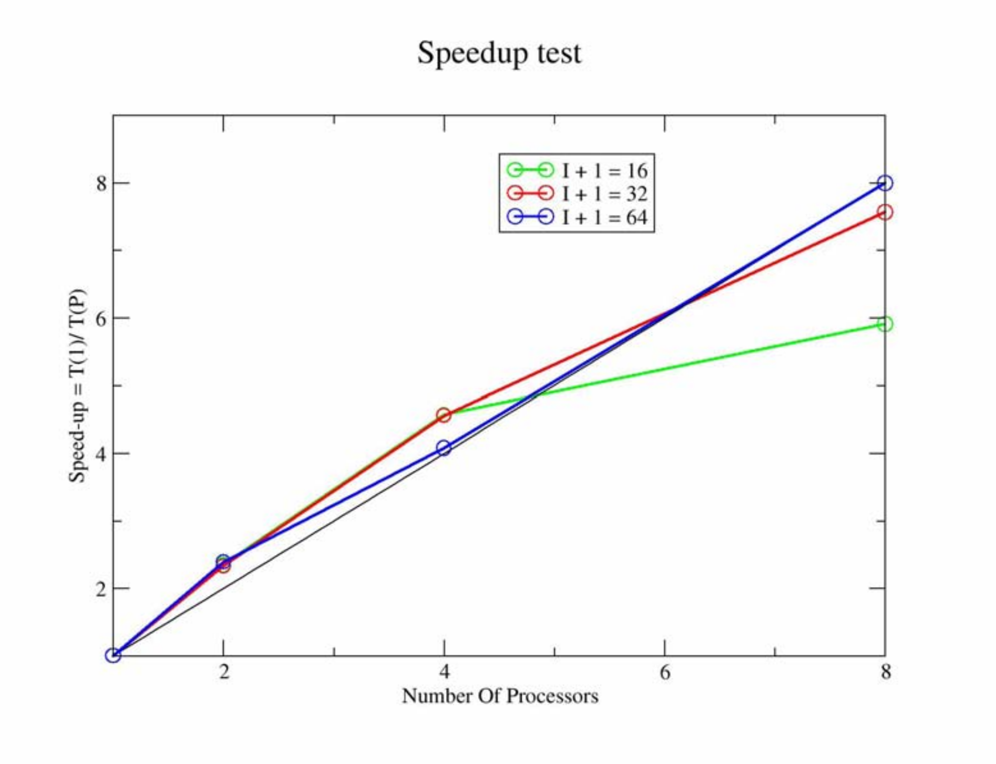


Figure 6.17: Speedup Test for parallel super-ensemble algorithm, for data set $0(\omega_\sigma = 0\%)$ and $\sigma_{n,s} = 1\% \times Z_{ref}$ in H_{X}, MC protocols as described in Section 5.2 with MC parameters as in Figure 6.5(a) and 6.6(a), with FE basis sizes $\mathcal{I} + 1 = 16, 32$, or 64 and parallel runs performed on $\mathcal{P} = 2, 4$ and 8 processors

Chapter 7

Summary and Conclusion

In summary, I have developed a new formalism and algorithm, referred to as the super-ensemble approach, for the simulation of ensembles of reaction kinetics models for chemical and biological systems. The super-ensemble approach is similar in spirit, and based on the same ideas, as the "standard" ensemble method, which was developed earlier [2, 20], for the kinetics model parameterization and prediction of biological networks from incomplete, noisy experimental data. However, at the computational level, the super-ensemble method eliminates the need for high-volume execution of conventional serial ODE solvers that have been used in the standard ensemble method. Instead, the super-ensemble method uses a Galerkin-type cost function, in conjunction with a variational Monte Carlo approach, to fully integrate the ODE solving task with the ensemble Monte Carlo exploration of the model parameter space.

For the case of a small model network for the kinetics of a simple enzyme catalysis process, I have demonstrated that the super-ensemble can correctly reconstruct kinetics model parameters, such as reaction rate coefficients or initial conditions. However, this must be understood with the caveat, that the super-ensemble can correctly reconstruct only those model parameters, (or quantities dependent on these model parameters) for which sufficient constraints are actually contained in the experimental input data. Just like any other scientific data analysis technique, the super-ensemble approach is subject to the "garbage-in-garbage-out" principle: it cannot extract any information *from* the data that is not contained *in* the data, as illustrated for the pair of poorly constrained rate coefficients in the simple enzyme model simulations. Yet, the ensemble does, in principle, provide us with the means to check which information is, or is not, contained in the data, and to quantify the degree of ignorance, given the data.

Unlike the standard ensemble method, the super-ensemble is easily and fully parallelizable by way of simple scalable time-domain decomposition methods. Parallelizability is achieved by using the Galerkin cost function approach instead of conventional strictly serial ODE solvers. In this thesis, I have implemented such a parallel algorithm as an MPI code and I have demonstrated its linear scaling with system size and processor number.

Another important aspect of the super-ensemble approach, which I have demonstrated in this thesis, is its potential utility as an ODE solver algorithm in conventional ODE solution applications. Unlike an intrinsically serial conventional ODE solver, the super-ensemble is highly parallelizable in the time domain. One should caution, however, that this computational advantage of parallelizability is “bought” at the expense of possibly having to perform a very large number of “sweeps” through the time-domain to minimize the Galerkin cost function H_{K}. By contrast, a conventional ODE solver has to sweep through the time domain only once to get a complete, highly accurate solution. A great deal of further development, optimization, and testing will be required to establish whether the super-ensemble can really be deployed as a competitive alternative in conventional ODE applications.

Further work will also be required to control and improve the Monte Carlo equilibration behavior of the super-ensemble approach. Possibly this could be done by exploiting a number of methods already developed for a problem that are computationally very similar to the super-ensemble: the simulation of continuum models for polymer systems. Such improvements in MC equilibration will likely also be critical for applications of the super-ensemble to substantially larger model networks describing “real-life” biological and chemical systems.

Bibliography

[1] Gene Amdahl. Validity of the single processor approach to achieving large-scale computing capabilities. *AFIPS Conference Proceedings*, 30:483–485, 1967.

[2] D. Battogtokh, D. K. Asch, M. E. Case, J. Arnold, and H.-B. Schttler. An ensemble method for identifying regulatory circuits with special reference to the *qa* gene cluster of *Neurospora crassa*. *PNAS*, 99(26):16904–16909, 2002.

[3] U.S. Bhalla and R. Iyengar. Emergent properties of networks of biological signaling pathways. *Science*, 283:381–387, 1999.

[4] Kevin S. Brown and James P. Sethna. Statistical mechanical approaches to models with many poorly known parameters. *Phys. Rev. E*, 68:021904–021913, 2003.

[5] Kevin Burrage. *Parallel and Sequential Methods for Ordinary Differential Equations*. Clarendon Press, 1995.

[6] Fergal P. Casey, Dan Baird, Qiyu Feng, Ryan N. Gutenkunst, Joshua J. Waterfall, Christopher R. Myers, Kevin S. Brown, Richard A. Cerione, and James P. Sethna. Optimal experimental design in an epidermal growth factor receptor signalling and down-regulation model. *IET Systems Biology*, 1:190–202, 2007.

[7] P. L. Ecuyer. Efficient and portable combined random number generators. *Communications of the ACM*, 31(6):742–774, 1988.

[8] C. A. J. Fletcher. *Computational Galerkin Methods*. Springer Verlag, 1984.

[9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[10] David P. Landau and Kurt Binder. *A Guide to Monte Carlo Simulations in Statistical Physic, Second Edition.* Cambridge University Press, 2005.

[11] D.A. Logan, A.L. Koch, W. Dong, J. Griffith, R. Nilsen, M.E. Case, H.B. Schuüttler, and J. Arnold. Genome-wide expression analysis of genetic networks in *Neurospora crassa*. *Bioinforation*, 1(10):390–395, 2007.

[12] J. D. Murray. *Lecture Notes on Nonlinear Differential Equation Models in Biology.* Clarendon Press, 1979.

[13] M.W.Covert, C.H.Schilling, I. Family, J.S.Edwards, I.I. Goryanin, E. Selkov, and B.O.Palsson. Metabolic modeling of microbial strains in silico. *Trends Biochem. Sci.*, 26:179–186, 2001.

[14] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing, Second Edition.* Cambridge University Press, 1992.

[15] Ramdas Ram-Mohan. *Finite Element and Boundary Element Applications in Quantum Mechanics.* Oxford University Press, 2002.

[16] Wolfgangl Reisig. *A Primer in Petri Net Design.* Springer-Verlag, 1992.

[17] Irwin H. Segel. *Enzyme Kinetics.* Wiley-Interscience, 1975.

[18] Irwin H. Segel. *Enzyme Kinetics: Behavior and Analysis of Rapid Equilibrium and Steady-State Enzyme Systems.* Wiley-Interscience, 1993.

[19] J.J. Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *PNAS*, 88:7328–7332, 1991.

[20] Yihai Yu, Wubei Dong, Cara Altimus, Xiaojia Tang, James Griffith, Melissa Morello, Lisa Dudek, Jonathan Arnold, and Heinz-Bernd Schttler. A genetic network for the clock of *Neurospora crassa*. *PNAS*, 104(8):2809–2814, 2007.