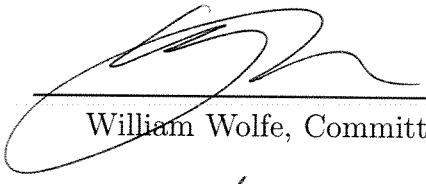
Modeling and Deterministic Simulation of Chemical Networks Under the Law of Mass Action

by
Ryan M. Brown

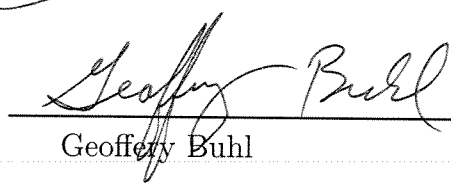
A thesis submitted in partial fulfillment of the requirements for the degree
of
Master Of Science
in
Computer Science
at
California State University Channel Islands

© 2008
Ryan M. Brown
ALL RIGHTS RESERVED

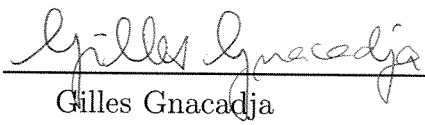
APPROVED FOR THE COMPUTER SCIENCE PROGRAM



William Wolfe, Committee Chairperson — Date 6/25/08

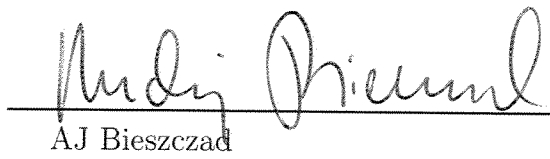


Geoffery Buhl — Date 6/25/08



Gilles Gnacadja — Date JUNE 25, 2008

APPROVED FOR THE UNIVERSITY



AJ Bieszczad — Date July 5th, 2008

To my sister, Amber Brown.

Acknowledgements

This dissertation could not have been written without the continuous confidence and encouragement of my mentor, Gilles Gnacadja; both throughout this work and during my internship at Amgen, his guidance has influenced my academic and professional life more than anyone else. Many thanks must also go to the members of my committee, Geoffery Buhl and Bill Wolfe. I would also like to thank Graham Matthews and for many helpful conversations. And to those who likewise endured long nights of studying throughout my time in graduate school, I would like to thank: Melanie Ivancic, David Mayorga, and Byron Sturtevant. Additionally, there are many family members and friends without whom this dissertation would not have been possible.

Abstract

Mass Action Kinetics refers to the supposition that the probability of chemical components reacting is proportional to their concentrations, with the rate constant taken as the constant of proportionality. We state a commonly used set-theoretic language for describing chemical networks and discuss the Law of Mass Action. Assuming a set of chemical substances react within a fluid mixture, isolated from the external universe, Mass Action Kinetics is used to model the time evolution of species within the system by nonlinear differential equations. Because these differential equations are considered stiff equations, comparative analysis of numerical methods shows that stiffness greatly determines the quality of computation. Specifically, a Taylor Series Method is comparable to both implicit and explicit Runge-Kutta Methods, though for stiff instances is inadequate. Examples demonstrating this contrast are provided, showing the need for caution when choosing a numerical integration techneque.

Contents

1 Preliminaries

1.1 Introduction

Systems of chemical reactions are common in many fields, notably in biochemistry, where reactions take place within a cell. These reactions are typically presented symbolically, indicating the proportions in which a set of initial chemicals interact to form other chemical substances. Chemistry literature notates a chemical reaction by a sum of initial chemical substances, with *stoichiometric coefficients* denoting the amount by which they exist in the chemical system, followed by a rightward *yields* arrow, and finally another sum of final chemicals.

A simple example of the notation used in chemistry literature to denote a chemical reaction is: $CH_4 + 2\,O_2 \longrightarrow CO_2 + H_2O$, which is refered to as the *chemical reaction equation*. While this bears resemblance to a standard mathematical equation, treatment as such often violates the chemical context.

When we refer to a system of chemical reactions, we mean a set of reactions that take place within the same environment. The reactions may inter-react or not, and when the final chemicals of a reaction are the initial chemicals of another, and vice-versa, we refer to the *reversible reaction* as the two collectively. By convention of the chemistry literature, we write a single chemical reaction equation, with a double-headed yield arrow. For example: $Fe_3O_4 + 4\,H_2 \rightleftharpoons 3\,Fe + 4\,H_2O$ is a reversible reaction, with individual reactions $Fe_3O_4 + 4\,H_2 \longrightarrow 3\,Fe + 4\,H_2O$ and $3\,Fe + 4\,H_2O \longrightarrow Fe_3O_4 + 4\,H_2$.

For each reaction there is an associated *rate* that governs the change in concentration of all related chemicals over time. The rates at which reactions occur can be difficult or even impossible to measure experimentally; they can vary over many orders of magnitude and thereby posses certain difficulties for chemists. In even mildly complicated systems there can be many coupled reactions with many such reaction constants, making the rigorous examination of the system very computationally intensive. Reaction rates serve as the premise for discussing the Law of Mass Action, which motivates the study of Mass Action Kinetics.

1.2 Notation and Terminology

We begin by introducing the the convention of multi-indices. Multi-index notation generalizes the concept of an integer index to an array of indices and allows for better explication of complicated formulae.

Definition 1. *A multi-index of dimension n is a vector $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ with components $\alpha_i \in \mathbb{Z}_{\geq 0}$.*

Notation 1. *For multi-indices $\vec{\alpha}, \vec{\beta} \in \mathbb{Z}^n_{\geq 0}$ and vector $\vec{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, let the following notational conventions hold:*

- $|\vec{\alpha}| = \sum_{i=1}^{n} \alpha_i$
- $\vec{\alpha}! = \prod_{i=1}^{n} \alpha_i!$
- $\vec{x}^{\vec{\alpha}} = \prod_{i=1}^{n} x_i^{\alpha_i}$
- $\mathbf{D}^{\vec{\alpha}} = \prod_{i=1}^{n} \mathbf{D}^{\alpha_i}$

Notation 2. *Raising x to the power of another vector is used in defining the operation of raising x to the power an $n \times r$ matrix $\mathbf{A}$. This gives an r dimensional vector*

$$x^{\mathbf{A}} = \begin{bmatrix} \vec{x}^{\vec{\alpha}_1} \\ \vdots \\ \vec{x}^{\vec{\alpha}_r} \end{bmatrix} \tag{1}$$

where $\vec{\alpha}_i$ is the i-th column of the matrix $\mathbf{A}$.

Multi-index notation allows for a convenient extension from univariate calculus formulae to the corresponding multivariate versions. The notational convention of multi-indices, along with the employment of multisets, eases the modeling process of chemical systems.

Definition 2. *Let X be a set. A multiset over X is a map $X \to \mathbb{Z}_{\geq 0}$. Conceptually, a multiset over X is a 'subset' of X, not necessarily proper, in which elements are possibly repeated.*

Notation 3. *A multiset A is usually denoted by the formal sum $A = \sum_{x \in X} \vartheta(x)x$ for some set X, where $\vartheta : X \to \mathbb{Z}_{\geq 0}$ gives the multiplicity of x in X.*

Example 1.1. A subset of set X is a multiset over X, the multiplicity map of which ranges into $\{0, 1\}$. In particular we have the empty multiset over X, which in the notation of formal summation, is denoted by 0.

Example 1.2. Let $X = \{a, b, c, d\}$. Then, $A = 2a + 3b + d$ is a multiset over X with the multiplicity map $\vartheta : X \to \mathbb{Z}_{\geq 0}$ enumeratively given by $\vartheta(a) = 2$, $\vartheta(b) = 3$, $\vartheta(c) = 0$, and $\vartheta(d) = 1$.

The use of multisets facilitates the mathematical formalism when discussing chemical reaction networks. The next section uses multisets to develop the needed mathematical language for describing the notation and verbiage used in chemistry literature.

2 Chemical Reaction Network Theory

Chemical reactions are formally defined as processes that interconvert a fixed collection of chemical substances. Herein, we consider reactions in a general context, existing together within a fluid mixture of a system, isolated from the external universe. If this assumption is violated, we say that the system is *open*, and otherwise we say that the system is *closed*. The chemical mixture is presumed sufficiently stirred, rendering the temperature of the system and distribution of concentrations uniform. These assumptions about the chemical system allow for global modeling of the system, rather than considering localized interacting subsystems.

For our purposes we refer to chemical substances as any *element*, *compound*, or *ion* without consdieration to its physical state. We must state that this approach is rather broad and may be too inclusive for describing certain other chemical phenomena.

Example 2.1. To illustrate a set of chemical reactions occurring in the well stirred mixture, consider the combustion of methane and the reversible formation of water:

$$\mathrm{CH_4} + 2\,\mathrm{O_2} \longrightarrow \mathrm{CO_2} + 2\,\mathrm{H_2O}$$
$$2\,\mathrm{H_2} + \mathrm{O_2} \rightleftharpoons 2\,\mathrm{H_2O}$$

The combustion reaction "converts" one molecule of methane and two molecules of oxygen to form one molecule of carbon-dioxide and two of water. Similarly, the water formation reaction converts two molecules of hydrogen and one of oxygen to produce two molecules of water, which is the "left to right" reaction $2\,\mathrm{H_2} + \mathrm{O_2} \longrightarrow 2\,\mathrm{H_2O}$; this is considered a *reversible* reaction, meaing that the formation of $\mathrm{H_2O}$ and its deformation back into $2\,\mathrm{H_2}$ and $\mathrm{O_2}$ occur simultaneously.

The formation of $\mathrm{H_2O}$ and its deformation back into $2\,\mathrm{H_2}$ and $\mathrm{O_2}$ are two reactions that are commonly observed in nature and used in a variety of scientific applications. Theorecically, any chemical reaction is reversible, though may not be in nature or in any laberatory setting. Accordingly, we consider only reversible reactions which may be demonstrated by experiment.

While we often consider reactions such as that of Example 2.1, generalizing to a set of chemical reactions lends to mathematical abstractions and systems modeling. More general still, we consider reactions that potentially involve an arbitrary number of chemical substances.

2.1 Mathematical Representation of Chemical Reactions

The depiction of chemical substances by their reaction equation does not possess a sufficient amount of "structure" for our modeling purposes. By considering sets of reactions as a whole, we ignore the role a particular chemical substance plays in each reaction. We henceforth employ multisets to construct a *chemical network*, which allows for better mathematical treatise.

Definition 3. *Given a set of chemical reactions, we define a chemical reaction network as a triple* $\breve{\mathscr{N}} = (\mathscr{S}, \mathscr{C}, \mathscr{R})$ *where*

- $\mathscr{S}$ *is the set of chemical species, assumed nonempty and finite;*
- $\mathscr{C}$ *denotes the set of complexes, having nonempty finite multisets on* $\mathscr{S}$ *such that for all* $x \in \mathscr{S}$*, the stoichiometric coefficient of* x *is its multiplicity in its complex.*
- $\mathscr{R}$ *is a binary relation on* $\mathscr{C}$ *with the property that for all* $y \in \mathscr{C}$*, there exists* $y' \in \mathscr{C}$ *such that either* $(y, y') \in \mathscr{R}$ *or* $(y', y) \in \mathscr{R}$ *holds.*

The term *species* is motivated by the literature that focuses specifically on chemical reaction networks. We adopt this convention and informally consider interpret species to mean the stateless chemical substances that participate in a reaction. This is motivated from the fact that a chemical equation, such as $\mathrm{CH_4} + 2\,\mathrm{O_2} \longrightarrow \mathrm{CO_2} + \mathrm{H_2O}$, may also be written as $\mathrm{CH_4} + 2\,\mathrm{O_2} + 0\,\mathrm{CO_2} + 0\,\mathrm{H_2O} \longrightarrow 0\,\mathrm{CH_4} + 0\,\mathrm{O_2} + \mathrm{CO_2} + \mathrm{H_2O}$; though this may seem like an arbitrary distinction, it shows that the reference to initial chemicals and final chemicals do not need to be explicit the writing of the chemical equation, except for chemistry considerations. For this example we have $\mathscr{S} = \{\mathrm{CH_4}, \mathrm{O_2}, \mathrm{CO_2}, \mathrm{H_2O}\}$ and the complex $\mathrm{CH_4} + 2\,\mathrm{O_2} + 0\,\mathrm{CO_2} + 0\,\mathrm{H_2O}$. We therefore often ignore the chemical equation and speak in more mathematical terms.

A key observation of the definition of a chemical network is that mass conservation is not included in the conditions. Specifically, there are no restrictions for reactions such as $\mathrm{A} \longrightarrow 2\,\mathrm{A}$ for some species A. Moreover, this definition does not exclude reactions such as $0 \longrightarrow \mathrm{A}$ or $\mathrm{A} \longrightarrow 0$, where 0 is commonly reffered to as "nothing" or the zero complex. Reactions that form species A from nothing, or destroy species A into nothing may seem absurd, but prove useful when defined properly.

Definition 4. *If a chemical reaction is of the form*

$$0 \longrightarrow \sum_{X \in \mathscr{S}} \mathrm{X}$$

or

$$\sum_{X \in \mathscr{S}} \mathrm{X} \longrightarrow 0$$

then it is said to contain the zero complex.

The utility of allowing for reactions to contain the zero complex proves useful for certain physical reactions. For instance, the in-flow of a substrate can be modeled with a pseudo-reaction of the form $0 \longrightarrow \mathrm{A}$; such reactions are typically open systems. Because many have focused on the application of chemical networks involving the zero complex to systems found in biochemistry, such as in [13], we contrastingly focus on closed chemical reaction networks that do not involve the zero complex because of its wide applicability to nonspecific reactions.

Under a particular ordering on $\mathscr{S}$ and $\mathscr{C}$, it is customary as in [1] to arrange the stoichiometric coefficeints in a $|\mathscr{S}| \times |\mathscr{R}|$ matrix, refered to as the *stoichiometric matrix*. Columns of the stoichiometric matrix correspond to chemical reactions within the network and rows correspond to chemical species. When a chemical reaction does not involve a particular chemical species, that stoichiometric coefficient is zero. We notate a stoichiometric matrix by $\mathcal{S}$, having elements $s_{i,j}$, where $0 \leq i \leq |\mathscr{S}|$ and $0 \leq j \leq |\mathscr{R}|$.

$$\mathcal{S} := \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,|\mathscr{R}|} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,|\mathscr{R}|} \\ \vdots & \vdots & \ddots & \vdots \\ s_{|\mathscr{S}|,1} & s_{|\mathscr{S}|,2} & \cdots & s_{|\mathscr{S}|,|\mathscr{R}|} \end{bmatrix}$$

Example 2.2. Consider the network given by the following reactions

$$\begin{aligned} \mathrm{A} &\rightleftharpoons \mathrm{B} \\ 0 &\longrightarrow \mathrm{C} \\ 2\,\mathrm{C} &\longrightarrow 3\,\mathrm{C} \\ \mathrm{D} &\longrightarrow 0 \end{aligned}$$

For this chemical network, the set of species is $\mathscr{S} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$, which implies $|\mathscr{S}| = 4$. Including the reversible reaction, the set $\mathscr{R}$ is such that $|\mathscr{R}| = 5$; under this enumeration of $\mathscr{S}$, the stoichiometric matrix for this network is

$$\mathcal{S} := \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 3 & -1 \end{bmatrix}$$

Example 2.2 serves to relate our use of the term *species* and the mathematical definition of the reaction set $\mathscr{R}$ with the allowance of the zero complex. Additionally, the construction of the stoichiometric matrix $\mathcal{S}$ will become useful later on.

Despite the wide applicability of chemical networks for modeling chemical systems, we refrain from incorporating any actual chemistry. Our construction of chemical networks depends only on set theoretic ideas, where elements of the set are taken as chemical species. Though this does not directly lead to describing the time evolution of species in a system, we have now sufficiently translated what is found in chemistry literature into a proper mathematical vocabulary.

2.2 The Law of Mass Action

The Law of Mass Action is typically introduced as the assumed governing phenomenon that drives chemical reactions. Literature on both the subjects of chemical reactions and the modeling of the time evolution of concentrations within a network is done so by this law. Because of its explanatory capabilities concerning concentrations of actual reactions, we take the law as

an axiom.

The origin of the Kinetic Law of Mass Action dates to Cato Maximilian Guldberg and Peter Waage, each of whom built on the previously proposed empirical rule that the rate of a chemical reaction is proportional to the product of the *active masses*; we take active mass to be synonymous with concentration for species in solution. The constant of proportionality characterizes the species' *affinity* to react. Interestingly, the Kinetic Law of Mass Action is shown to be a consequence of the empirically proposed Laws of Thermodynamics [17]. Because the term affinity is rather archain, we follow [17] and take mass action to include to notion of *chemical potential*.

Notation 4. *The concentration of some $x \in \mathscr{S}$ is denoted by $[x]$.*

Definition 5. *We refer to $\mathbb{R}^{|\mathscr{S}|}$ as the stoichiometric space of a chemical reation network. For any $x \in \mathscr{S}$, we have $[x] \in \mathbb{R}^{|\mathscr{S}|}$.*

Definition 6. *The constant of proportionality for active masses, is known as the rate constant and is denoted by k.*

Law of Mass Action. *For a chemical reaction network $\breve{\mathscr{N}}$ with $(y, y') \in \mathscr{R}$ and $[x] \in \mathbb{R}^{|\mathscr{S}|}$, let y_x denote the multiplicity of species x in its complex. If $\breve{\mathscr{N}}$ is dictated by the Law of Mass Action, then a reaction occurs at a rate proportional to the product of the species' concentrations. We may formulate the rate of a certain reaction occurring as given by the expression*

$$k_{(y,y')} \prod_{x \in \mathscr{S}} [x]^{y_x} \tag{2}$$

The Kinetic Law of Mass Action leads to the modern interpretation of chemical *equilibrium*, and thus founds the Equilibrium Law of Mass Action. A detailed analysis of the quantum statistical grounds for both Laws is provided by [17]. While the study of chemical equilibrium is often of great interest, analysis of chemical dynamics will be the primary focus herein.

2.3 Reaction Networks and Differential Equations

The Law of Mass Action serves as a vehicle for driving our model for describing the dynamics of a reaction network. While there are other formal models that take into account details of particular reaction types, we aim to have broad applicability and consider only *mass action systems*.

Definition 7. *A mass action system is a chemical reaction network $\breve{\mathscr{N}}$ under the Law of Mass Action, taken with a reaction rate coefficient map $k : \mathscr{R} \to \mathbb{R}_{>0}$ and is denoted by $\mathscr{N} = \{\mathscr{S}, \mathscr{C}, \mathscr{R}, k\}$.*

To describe the dynamics of a mass action system, literature on the subject compactly writes such the chemical reaction equations as

$$\sum_{\mathrm{A} \in \mathscr{S}} \alpha_A \mathrm{A} \xrightarrow{k} \sum_{\mathrm{A}' \in \mathscr{S}} \beta_{A'} \mathrm{A}' \tag{3}$$

denoting the idea that the reaction occurs at the rate of k. From the Law of Mass Action, we take this to mean that the rate of production of $x \in \mathscr{S}$ due only to contributions from this reaction is given by the expression

$$k \cdot y'_x \prod_{x \in \mathscr{S}} [x]^{y_x} \tag{4}$$

where $k := k\left((y, y')\right)$ for $(y, y') \in \mathscr{R}$. Similarly, the rate of destruction of $x \in \mathscr{S}$ is given by

$$-\, k \cdot y_x \prod_{x \in \mathscr{S}} [x]^{y_x} \tag{5}$$

We combine these two expressions to describe the net instantaneous rate of change in the chemical concentration of $x \in \mathscr{S}$ for this reaction:

$$\frac{d[x]}{dt} := k(y'_x - y_x) \prod_{x \in \mathscr{S}} [x]^{y_x} \tag{6}$$

By combining these two expressions for the formation and destruction of species, we arrive at an equation that describes the instantaneous rate of change in the chemical concentration. We apply this to the entire system and are thus able to depict the dynamics of the mass action system; this rate function is typically referred to as the *species formation function*.

Definition 8. *The species formation function for a mass action system $\mathscr{N}$, is the map $f : \mathbb{R}^{|\mathscr{S}|}_{\geq 0} \to \mathbb{R}^{|\mathscr{S}|}$ defined by*

$$f\left([x]\right) = \sum_{(y, y') \in \mathscr{R}} (y' - y) k_{y,y'} \prod_{x \in \mathscr{S}} [x]^{y_x} \tag{7}$$

Using Notation 1, the time evolution of a mass action system is given by the differential equation

$$\frac{d[x]}{dt} := f([x]) = \sum_{(y,y')\in\mathscr{R}} (y' - y) k_{y,y'} [x]^y \tag{8}$$

Example 2.3. To demonstrate the instantiation of a species formation function, we take an example from [15]. We consider the chemical system

$$\begin{gathered} \mathrm{A} \xrightarrow{k_1} \mathrm{B} \\ 2\,\mathrm{B} \xrightarrow{k_2} \mathrm{B} + \mathrm{C} \\ \mathrm{B} + \mathrm{C} \xrightarrow{k_3} \mathrm{A} + \mathrm{C} \end{gathered}$$

which has the vectorized species formation function

$$\begin{array}{ll} \dot{x}_1 = -k_1 x_1 + k_3 x_2 x_3 & x_1 := [\mathrm{A}] \\ \dot{x}_2 = k_1 x_1 - k_3 x_2 x_3 - k_2 x_2^2 & x_2 := [\mathrm{B}] \\ \dot{x}_3 = k_2 x_2^2 & x_3 := [\mathrm{C}] \end{array}$$

An alternate construction of the species formation function is by use of the stoichiometric matrix, $\mathcal{S}$:

$$f([x]) = \mathcal{S} \cdot \begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_{|\mathscr{R}|} \end{bmatrix} \cdot \Lambda \tag{9}$$

where $(y, y') \in \mathscr{R}$ and Λ is a vector having components $\prod_{x\in\mathscr{S}} [x]^{y_x}$.

Both forms of the species formation function are equivalent. Instantiation of the function from a set of reaction is preferably done by Equation 9, though literature mostly uses the form of Equation 7.

Because the species formation function is a *differential system*, we aim to solve the system in order to gain an explicit formula for the time evolution of concentrations of the species in our network. Doing so is often difficult or impossible and therefore we apply numerical methods of integration. This is

the main objective: determining which methods are best suited for numerically integrating the species formation function. Analysis of computational error, both cumulative and at any point of evaluation in an algorithm, is of critical concern; every method possesses certain properties for the consideration of error analysis. In the context of chemical reaction networks, we may exploit the chemical context for this purpose.

2.4 The Law of Concentration Conservation and Moieties

Like the Law of Mass Action, the Law of Conservation of Mass is an empirical rule, which states that the net mass of a closed system will remain constant, regardless of the processes acting within. The *conservation of concentrations* of species in solution may be regarded as a consequent implication. We may formalize this for any closed system by

$$\sum_{x \in \mathscr{S}} [x] = \grave{\gamma} \tag{10}$$

for some constant $\grave{\gamma} \in \mathbb{R}$, which may be taken as the sum of initial concentrations; this implies

$$\sum_{x \in \mathscr{S}} \frac{d[x]}{dt} = 0 \tag{11}$$

Equations 10 and 11 observe the conservation from a global perspective, considering all species in the network; it is possible to have additional conservation of concentration constraints within a system, which involve the species in some subset of $\mathscr{S}$.

Definition 9. *A conserved moiety, $\mathcal{M} \subseteq \mathscr{S}$, is a collection of chemical species that convert from one form to another but whose total amount never changes.*

The sum of the concentrations of the chemical species within a conserved moiety is a constant. We may conceptualize this in terms of the reactions within a network: that there are "sub-networks" that convert between chemical species, but none that create nor destroy chemical species. A common biological example of a conserved moiety is a chemical species that phosphorylates and dephosphorylates, but never synthesizes nor degrades, [1].

Definition 10. *Whenever*

$$\sum_{x \in \mathcal{M}} a_x \frac{d[x]}{dt} = 0$$

for some moiety $\mathcal{M}$ *and* $a_x \in \mathbb{R}$*, the equation is said to be a conservation constraint.*

Although conserved moieties are not the sole source of conservation constraints, which [1] notes, many of the conservation constraints in this dissertation derive from conserved moieties, and thus we liberally interchange the terms 'conservation constraint' or 'moiety constraint' in the sense of a moiety conservation constraint.

Just as we could add components of the species formation function to yield new relationships, we may add rows of the stoichiometric matrix. However, doing so no longer uniquely associates rows to the time evolution of a single species; for this reason, we cannot label the rows of the stoichiometric matrix by the enumeration of $\mathscr{S}$. To preserve proper labeling, we rightwardly augment the stoichiometric matrix by the $|\mathscr{S}| \times |\mathscr{S}|$ identity matrix prior to applying Gauss-Jordan elimination.

Theorem 1. *The number of moiety conservation constraints in a chemical reaction network is given by* $|\mathscr{S}| - \operatorname{rank}(\mathbf{W})$*, where* $\mathbf{W}$ *is the matrix given by rightwardly augmenting the stoichiometric matrix by the* $|\mathscr{S}| \times |\mathscr{S}|$ *identity matrix.*

Theorem 1 is stated in the discussion of [1] without much explanation; we give a sketch of a proof in order to validate the ability to determine how many conservation moieties exist in a chemical network before searching for any.

Proof. The maximal number of linearly independent rows of the rightwardly augmented stoichiometric matrix, $\mathbf{W}$, is $\operatorname{rank}(\mathbf{W})$. Since the stoichiometric matrix, $\mathbb{S}$, has $|\mathscr{S}|$ rows, so does $\mathbf{W}$. Therefore, the maximal number of linearly dependent rows of $\mathbf{W}$ is given by $|\mathscr{S}| - \operatorname{rank}(\mathbf{W})$. Because a linearly dependent row vector may be written as a linear combination of other rows, the stoichiometry of the dependent row is accounted for by the stoichiometry of finite sums of other rows. This implies a moiety conservation constraint on the dependent row vector in terms of the other reaction equations. □

Detection of conservation constraints is easily done as in [1], observing that the rows of the stoichiometric matrix are linearly dependent if and only if they represent a conservation constraint; this is largely due to the fact that row operations create combinations of components of the species formation function. If linear independence is observed, then these components form a linear combination that sums to zero, which implies that the concentrations sum to a constant. This gives a precise methodology for detecting conservation constraints within a chemical network via row reduction.

Determining linear independence may be done by applying Gauss-Jordan elimination to row-reduce the stoichiometric matrix. The Gauss-Jordan elimination process separates out the linearly dependent rows of the stoichiometric matrix, replacing those rows with zeros. Other row operations may lead to different conservation constraints, though all are chemically valid.

Example 2.4. Consider the chemical network previously given in Example 2.2, which has the set of species $\mathscr{S} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$. Under this enumeration and following [1], we rightwardly augment the stoichiometric matrix given by Example 2.2 with the 4×4 identity matrix:

$$\left[\begin{array}{ccccc|cccc} \text{-1} & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \text{-1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \text{-2} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 & \text{-1} & 0 & 0 & 0 & 1 \end{array}\right] \tag{12}$$

We then perform Gauss-Jordan elimination to produce the matrix.

$$\left[\begin{array}{ccccc|cccc} 1 & \text{-1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & \text{-2/3} & 0 & 0 & 1 & 2/3 \\ 0 & 0 & 0 & 1 & \text{-1/3} & 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right] \tag{13}$$

The leftmost portion of the reduced matrix has one row containing only zero entries, indicating that the system contains one conservation constraint, [1]. Let $x_1 := [A]$, $x_2 := [B]$, $x_3 := [C]$, and $x_4 := [D]$. We take the corresponding

rightmost portion of the row and multiply by the vector of concentrations.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = x_1 + x_2 \tag{14}$$

This means that $x_1+x_2 = \dot{\gamma}$ is constant throughout the reaction; equivalently, we have

$$\frac{dx_1}{dt} + \frac{dx_2}{dt} = 0$$

Now let k_1 and k_2 be the rate constants for the reactions

$$\mathrm{A} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathrm{B}$$

This conservation constraint is easily verified since the components of the species formation function

$$\frac{dx_1}{dt} = -k_1 x_1 + k_2 x_2$$

$$\frac{dx_2}{dt} = k_1 x_1 - k_2 x_2$$

sum to zero.

Example 2.5. Consider the chemical network $2\,\mathrm{A} + \mathrm{B} \longrightarrow \mathrm{C}$, which has the set of species $\mathscr{S} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and stoichiometric matrix

$$\mathcal{S} := \begin{bmatrix} -2 \\ -1 \\ 1 \end{bmatrix} \tag{15}$$

After rightwardly augmenting the matrix appropriately and then performing Gauss-Jordan elimination, we have the matrix

$$\left[\begin{array}{c|ccc} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \end{array}\right] \tag{16}$$

Since the second and third rows of the reduced stoichiometric matrix contain only zeros, we have determined the linear dependencies; as before, we take

the rightmost portion of the reduced augmented matrix and multiply by the vector of species.

$$\begin{bmatrix} 1 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1 + 2x_3 \tag{17}$$

$$\begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_2 + x_3 \tag{18}$$

so that $x_1 + 2x_3 = \grave{\gamma}_1$ and $x_2 + x_3 = \grave{\gamma}_2$ are moiety constraints.

Conservation of concentrations is useful within the scope of further understanding the relationships species play in a chemical network. Assuming that a chemical network is governed by the Law of Mass Action and scientifically follows the Conservation of Concentrations, Equations 10 and 11 provide a framework by which the error from numerical methods may scrutinized; additionally, we may search for conserved moieties to further scrutinize the error from numerical methods. Because this approach is applicable to any numerical method, conservation constraints play a key role in the comparative analysis of their error terms.

3 Numerical Methods Applied to the Species Formation Function

Using the species formation function as the model for the time evolution of the concentrations of chemical species, we may take the initial concetrations of the mass action system to be initial values of the differntial system. Thus we aim to explore numerical methods for solving these initial valued problems.

Notation 5. *When discussing numerical methods, we use the notation $x(t)$ to mean the that the concentration $[x]$ is a function of time. The initial concentrations are denoted by $x_0 := x(t_0)$ for the initial time t_0.*

Notation 6. *We write $\dot{x}$ to mean the first derivative of $x(t)$ with respect to time. This is analogous to Newton's notation of using superscript primes.*

The species formation function is autonomous, meaning $\dot{x}(t)$ is explicitly a function of x and satisfies the form

$$\dot{x} = f\left(x(t)\right) \quad x(t_0) = x_0 \tag{19}$$

where $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$. If we take the species formation function to be f, then $(t_0, x_0) \in \mathbb{R} \times \mathbb{R}^n$ is the initial concentration point, which must lie on the solution curve. Note that f is implicitly defined over $t \in \mathbb{R}$, and therefore may be explicitly defined over $\mathbb{R}^n$ to $\mathbb{R}^n$ only.

Definition 11. *A solution curve to an ordinary differential equation of the form 19 is a function $x : \mathbb{R} \to \mathbb{R}^n$ such that $\dot{x}(t) = f\left(x(t)\right)$ and $x(t_0) = x_0$.*

Due to the complexity of differential equations, explicit formulae for the solution function are rarely found. As an alternative, approximate solutions may be more easily determined; these are solutions which may hold accuracy only on certain intervals of t since they inherit instability from the approximation.

We discuss numerical methods that centralize around the *descritization* of the continuous interval $[a, b]$ of t, meaning that $[a, b]$ is partitioned into N subintervals of equal measure. The *steplength* or *stepsize*, denoted by h, is the discritization factor given by $h := (b - a)/N$; within a numerical method it is possible to adjust h, possibly providing for better results, though many

of the classical methods use a fixed steplength. While specialized methods have been developed, literature on chemical network modeling employs these classic methods.

3.1 Exact Solutions for Unimolecular Reactions

Unimolecular reactions with singular coefficients are those of the form $\mathrm{A} \longrightarrow \mathrm{B}$, where one chemical substance is directly converted into another; if we have a reaction involving zero-complex, then we still consider it unimolecular.

Because the coefficients of each species is 1, it is easy to find an exact solution the differential system given by Equation 7. This is simply because the equation assumes a linear form, which is solvable by standard calculus. While there are various ways to gain the exact solution, we introduce *matrix functions*, which allow for a convienent and compact notation.

Definition 12. *Let* $\mathbf{A}$ *be a real or complex* $n \times n$ *matrix. A function* g *on* $\mathbf{A}$ *is*

$$g(\mathbf{A}) := \sum_{\kappa=0}^{\infty} a_\kappa \mathbf{A}^\kappa \tag{20}$$

The canonical example of a matrix function is the *matrix exponential*. We define this specific matrix function by expounding upon the series expansion for the scalar exponential e^{at}.

Definition 13. *Let* $\mathbf{A}$ *be a real or complex* $n \times n$ *matrix. The matrix exponential is given by*

$$e^{\mathbf{A}t} := \sum_{\kappa=0}^{\infty} \frac{t^k}{\kappa!} \mathbf{A}^\kappa \tag{21}$$

where $e^0 = \mathbf{I}$*, which is well defined since the above series globally converges.*

Alternate definitions to the matrix exponential have been suggested, though all are equivalent; subsequent to the basic definition of the matrix exponential, [29] shows that many of the properties of the scalar exponential e^{at} translate into the matrix exponential.

Theorem 2. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be square constant matrices of common dimension. For* $r, s, t \in \mathbb{C}$*,*

- $e^{\mathbf{0}} = \mathbf{I}$ *where* $\mathbf{0}$ *is the zero matrix.*
- $e^{(\mathbf{A}+\mathbf{B})t} = e^{\mathbf{A}t}e^{\mathbf{B}t}$ *for multiplicatively commutative* $\mathbf{A}, \mathbf{B}$
- $e^{\mathbf{A}t}e^{\mathbf{A}s} = e^{\mathbf{A}(t+s)}$
- $e^{\mathbf{A}t}e^{-\mathbf{A}t} = \mathbf{I}$
- $\mathbf{D}\left(e^{\mathbf{A}t}\right) = \mathbf{A}e^{\mathbf{A}t}$

The proofs to each statement in Theorem 2 are fairly trivial; however we get some interesting and uselful implications from this theorem; $e^{\mathbf{A}t}$ is non-singular for all t, and therefore all columns are linearly independent solutions to the system $d\vec{x}/dt = \mathbf{A}x$. This is the argument that leads to the following result:

Theorem 3. *Let* $\mathbf{A}$ *be a square constant matrix and* $\dot{x} = d\vec{x}/dt$*. The general solution for the system* $\dot{x}(t) = \mathbf{A}\vec{x}(t)$ *with initial condition* $\vec{x}(t_0) = \vec{x}_0$ *is* $\vec{x}(t) = e^{\mathbf{A}(t-t_0)}\vec{x}_0$*. The columns of the matrix exponential* $e^{\mathbf{A}(t-t_0)}$ *form the fundamental solution set.*

Unfortunately, computation of $e^{\mathbf{A}t}$ is often rather difficult. Direct use of the series definition provides for poor results, as simple examples from [21] reveal. Many alternative methods have been devised, especially those found in [21], each of which either make use of diagonalizability, exploitation of the matrix exponential properties listed in Theorem 2, or various decompositions. Decompositions of the form $\mathbf{A} = \mathbf{SBS^{-1}}$ dually seek to find a well conditioned matrix $\mathbf{S}$ and a nearly diagonal $\mathbf{B}$, lending to easier computation of $e^{\mathbf{B}t}$, which may remain a challenge in practice.

While we may elect to perform computations manually, mathematical software provide the functionality to compute the matrix exponential. This lends well for large chemical systems. For a particular chemical reaction, where the rate constants are known, we may use Matlab®'s `expm` function to numerically compute $e^{\mathbf{A}t}$; this method employs *Padé approximation* as its computational engine. For symbolic computation of $e^{\mathbf{A}t}$, we elect to use Maple®, as the next example shows.

Example 3.1. For the elementary reaction

$$\mathrm{A} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathrm{B}$$

taken with the Law of Mass Action, let $\dot{x} := d\vec{x}/dt$; we have the differential system $\dot{x} = \mathbf{A}x$, where $x_1 := [\mathrm{A}]$ and $x_2 := [\mathrm{B}]$. We then have $\mathbf{A}$ given by

$$\mathbf{A} = \begin{bmatrix} -k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}$$

To use Maple®'s symbolic engine to compute the matrix exponential, we load the needed library: `with(LinearAlgebra)`. We need only now specify the matrix $\mathbf{A}$:

```
A := Matrix([[-k1, k2], [k1, -k2]])
```

which produces the output

```
    [-k1    k2 ]
A:= [          ]
    [k1     -k2]
```

To arrive at the final evaluation, we use `MatrixExponential(A, t)` and some trivial factorizations to produce the output

```
[ k2 + k1 exp(-(k1 + k2) t)       k2 (exp(-(k1 + k2) t) - 1) ]
[ -------------------------     - -------------------------- ]
[          k1 + k2                          k1 + k2          ]
[                                                            ]
[  k1 (exp(-(k1 + k2) t) - 1)     k1 + k2 exp(-(k1 + k2) t)  ]
[- -------------------------     -------------------------  ]
[           k1 + k2                        k1 + k2           ]
```

By defining an initial vector `x0:=<u,v>`, where `u` is the initial concentration of A_1 and `v` of A_2, we may then use the Maple® command `MatrixVectorMultiply(A,x0)` to effectively gain a general solution.

Given a chemical reaction whose differential system may be formulated in such a manner as to warrant its solvability by use of the matrix exponential,

we may choose a particular decomposition for better computational results. This may not be easy, since various conditions including diagonalizability, sparcity, etcetera may not initially be known or easy to determine. Nonetheless, many useful decompositions are found in [21] and [29], both of which provide deeper analysis of the matrix exponential, especially with a focus on Padé Approximation. As stated in [21], the decompositions are useful, but usually not sufficient in practice.

Because the method of matrix exponentials addresses only unimolecular reactions, a very small subset of all reaction types, it is not generally of great interest. Rather, we move on to discuss more applicable complex chemical networks and numerical methods for solving their instance of the species formation function.

3.2 Taylor Series Method

Efficient computation of transcendental and complicated functions is often done by the truncation of its *Taylor Series expansion*, which is a consequence of Taylor's Theorem. We begin discussion of its applicability in solving differential equations with initial values within a general context.

Theorem 4. *Let B be a ball in $\mathbb{R}^n$ centered at a point $c \in \mathbb{R}^n$. Let F be a real-valued function defined on the closure of B; if F has continuous partial derivatives of order $n+1$ at every point in B, then $\forall z \in B$*

$$F(z) = \sum_{|\alpha| \leq n} \frac{\mathbf{D}^\alpha F(c)}{\alpha!}(z-c)^\alpha + \sum_{|\alpha| \geq n+1} R_\alpha(z)(z-c)^\alpha$$

where $R_\alpha(z)$ is the remainder term, which in integral form is explicitly given by:

$$R_\alpha(z) = \sum_{|\alpha|=n+1} \frac{n+1}{\alpha!}(z-c)^\alpha \int_0^1 (1-t)^n (\mathbf{D}^\alpha F)(c+t(z-c))dt$$

and satisfies the inequality

$$|R_\alpha(z)| \leq \sup_{y \in B} \left| \frac{\mathbf{D}^\alpha F(y)}{\alpha!} \right|$$

Definition 14. *The truncated Taylor Series is said to be of order n if terms up to and including*

$$\frac{h^n \mathbf{D}^n(F)}{n!}$$

are within in the series expansion.

We have a dual application of Taylor's theorem: for a given function, we may approximate its values by evaluating a truncation on its Tayor series. This requires computing and evaluating the derivatives up to the truncation term. Alternatively we may be want to solve a differential equation, whereby we compute derivatives, which in conjunction with the differential system, constitutes all of the Taylor Series terms up to order $n + 1$.

Derivative computation is done in [4] by an iterative process, which is given without derivation. We consider a simpler methodology: let f_i be the a component of the species formation function for $1 \leq i \leq |\mathscr{S}|$ and $x := [x]$ for any $x \in \mathscr{S}$. By repeated use of the chain rule, we compute higher order derivatives by

$$\frac{d^{n+1} f_i}{dt^n} = \sum_{x \in \mathscr{S}} \frac{\partial^n f_i}{\partial x^n} \cdot \frac{d^n x}{dt^n} \tag{22}$$

for all $n \in \mathbb{Z}_{\geq 1}$, such that $n = 1$ is the species formation function. This approach may be a suboptimal process, but is easily implemented and has a clear derivation.

Since our aim is to solve instances of the species formation function, taking initial concentrations as the initial values to the system, we consider the second use of Taylor's Theorem, which constitutes *Taylor Series Method*. The primary disadvantage to this approach is the potential difficulty of computing and evaluating the needed derivatives; when using methods such as the Taylor Series, we are burdened with sufficiently minimizing the *truncation error* with the tedium of derivative computation.

3.3 Explicit Runge-Kutta Methods

A potential resolution to the disadvantage of the Taylor Series method is the use of Runge-Kutta methods. Runge-Kutta methods comprise a class of procedures that "sample" the solution space, gathering information of the

derivative. For this section we develop Runge-Kutta methods for nonautonomous equations, though we employ autonomous versions for chemical networks. Every definition and theorem hold in either case. Runge-Kutta methods may be partitioned into either explicit or implicit types, which sometimes give dramatically different results.

Definition 15. *Let $s \in \mathbb{Z}_{>1}$. For an ordinary differential equation $\dot{x}(t) = f(t, x)$ with initial value $x(t_0) = x_0$, the method*

$$\begin{aligned} k_1 &= f(t_0, x_0) \\ k_2 &= f(t_0 + c_2 h, x_0 + h a_{2,1} k_1) \\ k_3 &= f\left(t_0 + c_3 h, x_0 + h(a_{3,1} k_1 + a_{3,2} k_2)\right) \\ &\vdots \\ k_s &= f\left(t_0 + c_s h, x_0 + h(a_{s,1} k_1 + \cdots + as, s-1 k_{s-1})\right) \\ x_1 &= x_0 + h \textstyle\sum_{i=1}^{s} b_i k_i \end{aligned}$$

is refered to as an s-stange explicit Runge-Kutta method for $a_{2,1}, a_{3,1}, a_{3,2}, \ldots, a_{s,1}$, $b_1, \ldots, b_s$, and $c_2 \ldots, c_{s-1}$ all real coefficients.

The values c_i indicate the position, within the step, of the stage value. The matrix $\mathbf{A}$, having values $a_{i,j}$, indicates the dependence of the stages on the derivatives found at other stages. And values b_j are quadrature weights, which show how the final result depends on the derivates that are computed at various stages.

Definition 16. *A Runge-Kutta method has order p if for a given sufficiently smooth problem of the form 19,*

$$||x(t_0 + h) - x_1|| \leq K h^{p+1}$$

for some $K \in \mathbb{R}$, meaning that the "sampling" of the Taylor series occurs up to and including the term containing h^p.

Usually the c_i values satisfy the condition

$$c_i = \sum_{j=1}^{s} a_{i,j} \tag{23}$$

which professes that all points where f is evaluated are first order approximations to the solution. By the construction of each b_i, we empose the condition $\sum_{i \leq s} b_i = 1$.

It has become customary since the work of [6] to represent the Runge-Kutta coefficients in a *Butcher tableau*:

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{2,1} & & & & \\ c_3 & a_{3,1} & a_{3,2} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

If we define $b^T := [b_1, b_2, \ldots, b_s,]$, then we may compactly write the Runge-Kutta methods in a *Butcher matrix*:

$$\begin{array}{c|c} c & \mathbf{A} \\ \hline & b^T \end{array} \tag{24}$$

with $c, b \in \mathbb{R}^s$ and $\mathbf{A} \in \mathbb{R}^{s \times s}$ where $\mathbf{A}$ is lower triangular having zeros along its diagonal. From Equation 23, we have the row sum of $\mathbf{A}$ equal to the corresponding value of c.

Much work has been done in determining the coefficients of 4-stage Runge-Kutta methods, such that they remain of order 4. In order to reproduce these results, we compute derivatives of $x_1 := x_1(h)$ at $h = 0$. By use of Equation 23, [14] derives the equations

$$\begin{array}{l} \sum_i b_i = b_1 + b_2 + b_3 + b_4 = 1 \\ \sum_i b_i c_i = b_2 c_2 + b_3 c_3 + b_4 c_4 = \frac{1}{2} \\ \sum_i b_i c_i^2 = b_2 c_2^2 + b_3 c_3^2 + b_4 c_4^2 = \frac{1}{3} \\ \sum_{i,j} a_{i,j} b_i c_j = a_{3,2} b_3 c_2 + a_{4,2} b_4 c_2 + a_{4,3} b_4 c_3 = \frac{1}{4} \\ \sum_{i,j} a_{i,j} b_i c_i c_j = a_{3,2} b_3 c_3 c_2 + a_{4,2} b_4 c_4 c_2 + a_{4,3} b_4 c_4 c_3 = \frac{1}{8} \\ \sum_{i,j} a_{i,j} b_i c_j^2 = a_{3,2} b_3 c_2^2 + a_{4,2} b_4 c_2^2 + a_{4,3} b_4 c_3^2 = \frac{1}{12} \\ \sum_{i,j,\ell} a_{i,j} a_{j,\ell} b_i c_\ell = a_{4,3} a_{3,2} b_4 c_2 = \frac{1}{24} \end{array} \tag{25}$$

Historically, analysis of these coefficients is done by certain "simplifying conditions," which were first done in [6]. These assumptions apply to higher order cases as well.

Lemma 5. *Suppose that for* $j = 1, \ldots, s$

$$\sum_{i=j+1}^{s} a_{i,j} = b_j(1 - c_j) \tag{26}$$

Then the first three equations from Equation 25 imply the last three.

Lemma 6. *Let* U *and* V *be* 3×3 *matrices such that*

$$UV := \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

satisfying the condition that $ad - bc \neq 0$. *Then for* $\nu = [0, 0, 1]^T$, *either* $V\nu = 0$ *or* $U^T\nu = 0$.

Proof. If $\det(U) \neq 0$, then $UV\nu = 0$, which implies $V\nu = 0$. Otherwise, if $\det(U) = 0$, then there exists some nonzero vector $\vec{x} := [x_1, x_2, x_3]^T$ such that $U^T\vec{x} = 0$; therefore, $V^TU^T\vec{x} = 0$. But, by the construction of UV, $\vec{x}$ must be a multiple of ν. □

Theorem 7. *For* $s = 4$, *Equations 26 and 23 imply Equation 25.*

Proof. Define $d_j := \sum_i a_{i,j}b_i - b_j(1 - c_j)$ for $j = 1, \ldots, 4$. To show that $d_j = 0$, let

$$U := \begin{bmatrix} b_2 & b_3 & b_4 \\ b_2c_2 & b_3c_3 & b_4c_4 \\ d_2 & d_3 & d_4 \end{bmatrix}$$

and

$$V := \begin{bmatrix} c_2 & c_2^2 & \sum_j a_{2,j}c_j - \frac{c_2^2}{2} \\ c_3 & c_3^2 & \sum_j a_{3,j}c_j - \frac{c_3^2}{2} \\ c_4 & c_4^2 & \sum_j a_{4,j}c_j - \frac{c_4^2}{2} \end{bmatrix}$$

By multiplying U and V, followed by employment of Equation 25, we have

$$UV = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{4} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The last column of V cannot be zero, since $c_1 = 0$ implies $\sum_j a_{2,j}c_j - \frac{c_2^2}{2} \neq 0$ by Equation 25. The result $d_2 = d_3 = d_4 = 0$ is given by Theorem 6, from which we conclude that $d_1 = 0$ since $d_1 + d_2 + d_3 + d_4 = 0$ by first and second equations of Equation 25. □

From Theorems 26 and 7 we simplify our conditions on the coefficients by the following theorem.

Theorem 8. *Under the assumption from Equation 23, the equations of 25 are equivalent to*

$$\begin{aligned}
&b_1 + b_2 + b_3 + b_4 = 1 \\
&b_2c_2 + b_3c_3 + b_4c_4 = \tfrac{1}{2} \\
&b_2c_2^2 + b_3c_3^2 + b_4c_4^2 = \tfrac{1}{3} \\
&b_2c_2^3 + b_3c_3^3 + b_4c_4^3 = \tfrac{1}{4} \\
&a_{3,2}b_3c_3c_2 + a_{4,2}b_4c_4c_2 + a_{4,3}b_4c_4c_3 = \tfrac{1}{8} \\
&a_{3,2}b_3 + a_{4,2}b_4 = b_2(1 - c_2) \\
&a_{4,3}b_4 = b_3(1 - c_3) \\
&b_4(1 - c_4) = 0
\end{aligned} \tag{27}$$

It follows that $b_3b_4c_2(1 - c_3) \neq 0$ and $c_4 = 1$, as in [14]. Many particular solutions to the system in Theorem 8 have been found, though the canonized one is

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

which is usually what is meant when refering to "the" Runge-Kutta method. We interpret each b_i and c_i as the coefficients of a fourth order quadrature formula such that $c_1 = 0$ and $c_4 = 1$.

While we mainly consider a fourth order Runge-Kutta method, a great amount of effort has been devoted to the understanding of the conditions that must be satisfied to achieve a particular order; these are commonly refered to as the "order conditions." The structure of order conditions originates in [6]

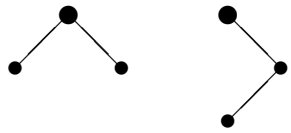


Figure 1: The two trees having three nodes, $\tau_{3;1}$ and $\tau_{3;2}$, respectively

and has been expounded upon in many sources, [14] and [19]. The theory for deriving the order conditions of a Runge-Kutta method is based on analyses of rooted trees.

Definition 17. *A rooted tree of n-th order is a set of n nodes joined by lines where each branch originates from a common node, the root, and no two branches are not allowed to grow together again afterward.*

Notation 7. *A rooted tree will be named with a notation $\tau_{i;j}$ where the first index, i, states the amount of nodes and index j is an internal enumeration in the class of trees with i nodes.*

The simplest tree is $\tau_{1,1}$ having the root as its only node and no braches. Sometimes it is useful to refer to the tree having no nodes, which is denoted by $\tau_{0;0}$. Figure 1 shows the two trees that have three nodes.

We employ the concept of *grafting* trees together by constructing a new tree where each node adjuct to the root is a root to an original tree. Consequently, every tree with p nodes can be constructed by taking trees with cumulative order $p-1$ and grafting them onto a new root. Using this point of view we can notate every tree as $\tau = [\tau_{i;j}, \tau_{k;l}, \ldots, \tau_{m;n}]$ where $\tau_{i;j}, \tau_{k;l}, \ldots, \tau_{m;n}$ are the trees that are grafted to a new root to form τ; if some "subtree," $\tau_{i;j}$, appears in the grafting n times, the we replace all $\tau_{i;j}$ in this notation with a single $\tau_{i;j}^{n}$.

Definition 18. *Let τ be a rooted tree. We define certain functions on rooted trees.*

1. *The order of τ is given by*

$$r(\tau) := 1 + \sum_{\tau_{i;j}^{\iota} \in \tau} \iota \cdot \tau_{i;j}$$

2. *The symmetry of τ is given by*

$$\sigma(\tau) := \prod_{\tau_{i;j}^{\iota} \in \tau} \iota! \cdot \sigma(\tau_{i;j})^{\iota}$$

Tree	$r(\tau)$	$\sigma(\tau)$	$\Psi(\tau)$
$\tau_{1,1}$	1	1	1
$\tau_{2,1}$	2	1	2
$\tau_{3,1}$	3	2	3
$\tau_{3,2}$	3	1	6
$\tau_{4,1}$	4	6	4
$\tau_{4,2}$	4	1	8
$\tau_{4,3}$	4	2	12
$\tau_{4,4}$	4	1	24

Figure 2: Functions $r(\tau)$, $\sigma(\tau)$, $\gamma(\tau)$ for rooted trees of order 1-4

3. *The density of* τ *is given by*

$$\Psi(\tau) := r(\tau) \cdot \prod_{\tau_{i;j}^{\iota} \in \tau} \gamma(\tau_{i;j})^{\iota}$$

4. *The elementary weights of* τ *are given by*

$$\beth(\tau) := \bigodot_{\tau_{i;j}^{\iota} \in \tau} A \cdot (\beth(\tau_{i;j}))^{\iota}$$

where $\bigodot$ *is component-wise multiplication*

5. *Let* $\varpi = \sum_{\tau_{i;j}^{\iota} \in \tau} \iota$ *The elementary differential of* τ *is given by*

$$\gimel(\tau) := f^{(\varpi)} \cdot \left\{ \gimel(\tau_{i;j}) \text{ with multiplicity } \iota | \tau_{i;j}^{\iota} \in \tau \right\}$$

with $r(\tau_{1;1}) = \sigma(\tau_{1;1}) = \Psi(\tau_{1;1}) = 1$ *and* $\gimel(\tau_{1;1}) = f$.

According to [9], paraphrasing [6], the local truncation error $x(t_{k+1}) - x_{k+1}$ is given by

$$x(t_{k+1}) - x_{k+1} = \sum_{i=1}^{\infty} \sum_{\tau \in T_i} h^i \cdot \frac{1}{\sigma(\tau)} \left(b^T \cdot \beth(\tau) - \frac{1}{\Psi(\tau)} \right) \cdot \gimel(\tau) \tag{28}$$

where T_i is the set of rooted trees of order i, σ and γ are integer-valued functions of τ, $\gimel$ is an elementary differential, and $\beth(\tau) \in \mathbb{R}$ is is a certain

composition of A, b, c, with a form that depends only on τ.

From [9], a Runge-Kutta method is of order p if and only if

$$\beth(\tau) := \frac{1}{\sigma(\tau)} \left(b^T \cdot \daleth(\tau) - \frac{1}{\Psi(\tau)} \right) = 0$$

for every $\tau \in T_i$ and $i \in \mathbb{N}_p$. This relation defines the order conditions, which are linear in the components of b and nonlinear in the components of A and c, which consequently relates them to the rooted trees.

Example 3.2. To derive the order conditions for a first order Runge-Kutta method, we have

$$\beth(\tau_{1;1}) := \frac{1}{\sigma(\tau_{1;1})} \left(b^T \cdot \daleth(\tau_{1;1}) - \frac{1}{\Psi(\tau_{1;1})} \right) = \sum_i b_i - 1 = 0$$

Example 3.3. To derive the order conditions for a second order Runge-Kutta method, we have the tree structure $\tau_{2;1} = [\tau_{1;1}]$, and therefore

$$\beth(\tau_{2;1}) := \frac{1}{\sigma(\tau_{2;1})} \left(b^T \cdot \daleth(\tau_{2;1}) - \frac{1}{\Psi(\tau_{2;1})} \right) = b^T \cdot c - \frac{1}{2} = 0$$

Example 3.4. To derive the order conditions for a third order Runge-Kutta method, we have the tree structures $\tau_{3;2} = [\tau_{1;1}^2]$ and $\tau_{3;1} = [\tau_{2;1}]$, and therefore we have system of equations

$$\beth(\tau_{3;2}) := \frac{1}{\sigma(\tau_{3;2})} \left(b^T \cdot \daleth(\tau_{3;2}) - \frac{1}{\Psi(\tau_{3;2})} \right) = b^T \cdot A \cdot c - \frac{1}{6} = 0$$

$$\beth(\tau_{3;1}) := \frac{1}{\sigma(\tau_{3;1})} \left(b^T \cdot \daleth(\tau_{3;1}) - \frac{1}{\Psi(\tau_{3;1})} \right) = \frac{1}{2} \left(b^T c^2 - \frac{1}{3} \right) = 0$$

There are four rooted trees of fourth order, as depicted in Figure 3; subsequently, we can continue to produce all the required order conditions for a Runge-Kutta method to ensure that we have order 4, which are given in Equation 25.

Clearly the amount of conditions that must be satisfied when increasing a Runge-Kutta method's order grows at a rate faster than that of the order. Unfortunately, we cannot continue the process of order condition derivation while ensuring the accuracy of the method.

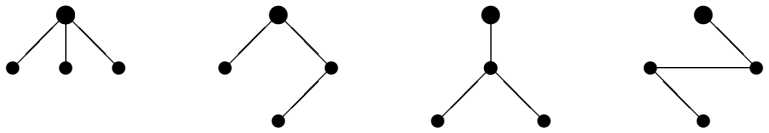


Figure 3: The four trees having four nodes

Order p	1	2	3	4	5	6	7	8	9	10
Amount of conditions	1	2	4	8	17	37	85	200	486	1205

Figure 4: Total amount of conditions to achieve order p

Theorem 9. *For $p \geq 5$, no explicit Runge-Kutta method exists of order p with $s = p$ stages.*

Proof. Consider the case $p = 5 = s$ and define the matrices U and V by

$$U = \begin{bmatrix} \sum_i b_i a_{i,2} & \sum_i b_i a_{i,3} & \sum_i b_i a_{i,4} \\ \sum_i b_i a_{i,2} c_2 & \sum_i b_i a_{i,3} c_3 & \sum_i b_i a_{i,4} c_4 \\ g_2 & g_3 & g_4 \end{bmatrix}$$

$$V = \begin{bmatrix} c_2 & c_2^2 & \sum_j a_{2,j} c_j - \frac{c_2^2}{2} \\ c_3 & c_3^2 & \sum_j a_{3,j} c_j - \frac{c_3^2}{2} \\ c_4 & c_4^2 & \sum_j a_{4,j} c_j - \frac{c_4^2}{2} \end{bmatrix}$$

where $g_k = \sum_{i,j} b_i a_{i,j} - \frac{1}{2} \sum_i b_i a_{i,k} (1 - c_k)$. From [14], the order conditions for a fifth order Runge-Kutta method imply

$$UV = \begin{bmatrix} \frac{1}{6} & \frac{1}{12} & 0 \\ \frac{1}{12} & \frac{1}{20} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

From Lemma 6 we have $g_4 = 0$ and consequently $c_4 = 1$. So,

$$g_j = \left(\sum_i b_i a_{i,j} - b_j (1 - c_j) \right) (c_j - c_5)$$

Since UV is the same as above, it follows that $c_4 = c_5$, and consequently $c_5 = 1$. Because $2 \leq k < j < i$, we have

$$\sum_{i,j,k} b_i (1 - c_i) a_{i,j} a_{j,k} c_k = 0$$

Amount of stages	1	2	3	4	5	6	7	8	9
Max obtainable order	1	2	3	4	4	5	6	6	7

Figure 5: Order barriers for various stages

From [14], by expanding this sum and using two fifth-order conditions, the expression should be 120, which is a contradiction. The inductive step is provided by [14] using the same techniques. □

The result of Theorem 9 is refered to as the *Butcher barrier* of a Runge-Kutta method. From [9], we have Figure 5, which shows the order barriers for high-stage methods.

According to the theory of order conditions for Runge-Kutta methods by rooted trees, a one-to-one relation can be defined between the set of order p conditions and the set of rooted trees with p nodes. Hence, the formation of the trees with p nodes can lead us to the corresponding order conditions of order p. This is the popular approach to understanding the comparative properties of Runge-Kutta methods; because of the Butcher barrier and because the formation of expressions for order conditions is a tedious task, even when we follow the theory using rooted trees, it is clear why fourth order methods are prefered.

3.4 Implicit Runge-Kutta Methods

Implicit Runge-Kutaa methods are natural extensions of the previously discussed explicit methods. The same definition of order holds for implicit methods and similarly to explicit methods, their order conditions are derived in the same manner and an equivalent notion of rooted trees exists.

Definition 19. *For $s \in \mathbb{N}$ and $i, j \in \mathbb{N}_s$, the general "s-stage" Runge-Kutta method is defined by*

$$k_i = f\left(t_0 + c_i h, x_0 + h\sum_{j=1}^{s} a_{i,j}k_j\right) \tag{29}$$

$$x_1 = x_0 + h\sum_{i=1}^{s} b_i k_i$$

for real numbers $a_{i,j}$, b_i, and c_i.

When $a_{i,j} = 0$ for $i \leq j$ we have an *explicit* method. If $a_{i,j} = 0$ and at least one $a_{i,i} \neq 0$, the method is commonly said to be*diagonally implicit*. Additionally, a *singly diagonally implicit* method is such that all diagonal elements are identical, meaning $a_{i,i} = \gamma$ for $i = 1, \ldots, s$; contrary to [14], we collectively speak of implicit Runge-Kutta methods.

The natural extension of explict Runge-Kutta methods to implicit ones is best seen in the arrangement of the Butcher tableu, as in 24, since we no longer require that the matrix $\mathbf{A}$ be lower triangular.

Example 3.5. The classicl *backward Euler method* is perhaps the simplest implicit Runge-Kutta method, which has Butcher tableu

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

While we have examined explicit Runge-Kutta methods in detail, we did not address the question: does a Runge-Kutta method approximate the actual solution to a differential equation? The answer to this is in Theorem 10, which shows the existence and uniqueness of a numerical solution; this is a general theorem for all implicit methods, for which $s \cdot n$ unknowns must be determined, if each k_i is of dimension n.

Theorem 10. *Let $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ be continuous and satisfy a Lipschitz condition with constant L. If*

$$h < \frac{1}{L \max_i \sum_j |a_{i,j}|}$$

then there exists a unique solution of 29, which may be obtained by iteration. If $f(t, x)$ is p times continuously differentiable, then the functions k_i as functions of h are also in C^p.

Proof. We show the existence by iteration

$$k_i^{(m+1)} = f\left(t_0 + c_i h, x_0 + h \sum_{j=1}^{s} a_{i,j} k_j^{(m)}\right)$$

Define $K \in \mathbb{R}^{sn}$ by $K = [k_1, k_2, \ldots, k_s]$. Let $\|K\| = \max_i (\|k_i\|)$; equation 29 may therefore be rewritten as $K := F(K)$ where

$$F_i = f\left(t_0 + c_i h, x_0 + h\sum_{j=1}^{s} a_{i,j} k_j\right)$$

By the Lipschitz condition and repeated use of the triangle inequality

$$\|F(K_1) - F(K_2)\| \leq hL \max_{i \in \mathbb{N}_s} \sum_{j=1}^{s} a_{i,j} \cdot \|K_1 - K_2\|$$

which by our assumption is a contraction. The Contraction Mapping Principle ensures the existence and uniqueness of the solution, as well as the convergence of the fixed point iterations, [14]. The differentiability is ensured by the Implicit Function Theorem, as expounded upon in [14]. □

A potential difficulty arrises when we consider *stiff systems*, which invalidates our suppostion that

$$h < \frac{1}{L \max_i \sum_j |a_{i,j}|}$$

is true. An all-inclusive existence theorem is provided in [15], but for brevity is herein omitted.

We note that the k_i equations are not able to be evaluated successively, since Equation 29 is a system of implicit equation for the determination of k_i. This typically results in poor perfomance in computation time and space; while this is a certain cost of implicit Runge-Kutta methods, the following theorems provide for certain advantages in their implementations.

Theorem 11. *If the following conditions are satisfied*

$$B(p) := \sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q} \qquad q \in \mathbb{N}_p$$

$$C(\xi) := \sum_{j=1}^{s} a_{i,j} c_J^{q-1} = \frac{c_i^q}{q} \qquad i \in \mathbb{N}_s,\ q \in \mathbb{N}_\xi$$

$$D(\zeta) := \sum_{i=1}^{s} a_{1,j} b_i c_i^{q-1} = \frac{b_j}{q}\left(1 - c_j^q\right) \quad j \in \mathbb{N}_s,\ q \in \mathbb{N}_\zeta$$

such that $p \leq 2\xi + 2$ *and* $p \leq \xi + \zeta + 1$*, then the method is of order* p*.*

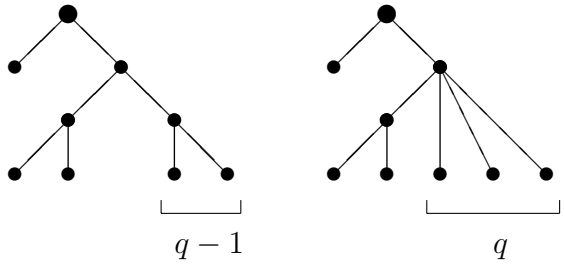


Figure 6: Reduction with $C(\xi)$

The assumption $C\left(\xi\right)$ implies that the two trees depicted in Figure 6 have identitcal order conditions for $q \leq \xi$. In contrast to high-order explicit Runge-Kutta methods, we do not require conditions such as $b_2 = 0$, as in [14].

The assumption $D\left(\zeta\right)$ is an extension of conditions imposed on fourth order explicity methods, which is expounded upon in [14].

Considering computations of implicit Runge-Kutta methods under these simplifying assuptions, we have a critical interprentation of $C\left(\xi\right)$.

Theorem 12. *The assumption of $C\left(\xi\right)$ implies that the internal stages*

$$g_i = x_0 + h\sum_{j=1}^{s} a_{i,j}k_j \tag{30}$$

where $k_j = f(t_0 + c_j h, g_j)$ satisfy for all $i \in \mathbb{N}_s$

$$g_i = x(t_0 + c_i h) = O(h^{\xi+1}) \tag{31}$$

Proof. Because of $C\left(\xi\right)$, the exact solution satisfies the Taylor expansion

$$x(t_0 + c_i h) = x_0 + h\sum_{j=1}^{s} a_{i,j}\dot{x}(t_0 + c_j h) + O(h^{\xi+1}) \tag{32}$$

Subtracting Equation 32 from 30 yeilds

$$g_i - x(t_0 + c_i h) = h\sum_{j=1}^{s} a_{i,j} f\left(t_0 + c_j h, g_j\right) - a_{i,j} f\left(t_0 + c_j h, x(t_0 + c_j h)\right) + O(h^{\xi+1})$$

If we assume Lipschitz continuity of f, then we derive Equation 31. □

Under the assumptions of $B\left(p\right)$, $C\left(\xi\right)$, and $D\left(\zeta\right)$, we are able to obtain methods of order $2s$ for any value s. This is done by considering $B\left(2s\right)$ and $C\left(s\right)$, which implies $D\left(s\right)$. We apply Theorem 11 with the values $p = 2s$, $\xi = s$, and $\zeta = s$. Therefore, any method obtained in this fashion is of order

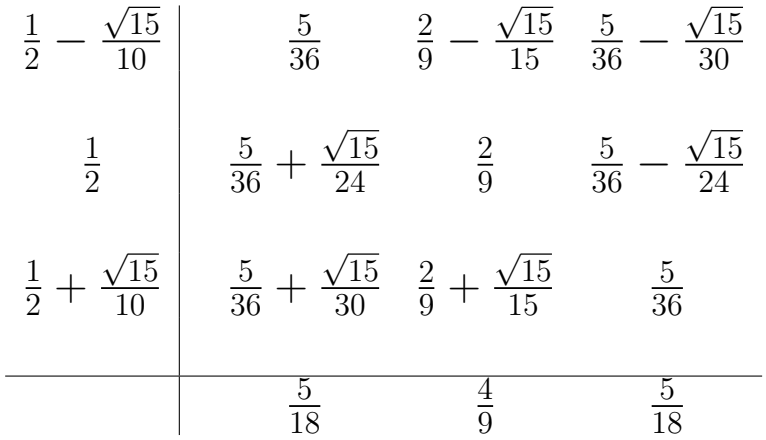


Figure 7: Bucher coefficients for implicit Runge-Kutta method of order 6

$2s$. From [6], Butcher coefficients are given in Figure 7 for $s = 3$ and in Figure 8 for $s = 4$.

The main difficulty when implementing a Runge-Kutta method is the balancing of accuracy needs, which is obtained from large order and many stages, finding vectors c and b, as well as each element of $\mathbf{A}$ such that the order conditions hold. Theorem 11 allows for more possible values to be found for order conditions, while ensuring higher accuracy.

$\frac{1}{2}-\varpi_2$	ϖ_1	$\varpi_1'-\varpi_3+\varpi_4'$	$\varpi_1'-\varpi_3-\varpi_4'$	$\varpi_1-\varpi_5$
$\frac{1}{2}-\varpi_2'$	$\varpi_1-\varpi_3'+\varpi_4$	ϖ_1'	$\varpi_1'-\varpi_5'$	$\varpi_1-\varpi_3'-\varpi_4$
$\frac{1}{2}+\varpi_2'$	$\varpi_1+\varpi_3'+\varpi_4$	$\varpi_1'+\varpi_5'$	ϖ_1'	$\varpi_1+\varpi_3'-\varpi_4$
$\frac{1}{2}+\varpi_2$	$\varpi_1+\varpi_5$	$\varpi_1'+\varpi_3+\varpi_4'$	$\varpi_1'+\varpi_3-\varpi_4'$	ϖ_1
	$2\varpi_1$	$2\varpi_1'$	$2\varpi_1'$	$2\varpi_1$

$$\varpi_1=\tfrac{1}{8}-\tfrac{\sqrt{30}}{144} \qquad \varpi_1'=\tfrac{1}{8}+\tfrac{\sqrt{30}}{144}$$

$$\varpi_2=\tfrac{1}{2}\sqrt{\tfrac{15+2\sqrt{30}}{35}} \qquad \varpi_2'=\tfrac{1}{2}\sqrt{\tfrac{15-2\sqrt{30}}{35}}$$

$$\varpi_3=\varpi_2\left(\tfrac{1}{6}+\tfrac{\sqrt{30}}{24}\right) \qquad \varpi_3'=\varpi_2'\left(\tfrac{1}{6}-\tfrac{\sqrt{30}}{24}\right)$$

$$\varpi_4=\varpi_2\left(\tfrac{1}{21}+\tfrac{5\sqrt{30}}{168}\right) \qquad \varpi_4'=\varpi_2'\left(\tfrac{1}{21}-\tfrac{5\sqrt{30}}{168}\right)$$

$$\varpi_5=\varpi_2-2\varpi_3 \qquad \varpi_5'=\varpi_2'-2\varpi_3'$$

Figure 8: Bucher coefficients for implicit Runge-Kutta method of order 8

4 Stiffness

A natural question to ask is: given various numerical methods for approximating the time evolution of instances of the species formation function, is the choice for implementation arbitrary? Because the species formation function is an example of what the literature on numerical methods terms *stiff equations* or *stiff systems*, the choice of method is not arbitrary. What is actually meant by *stiffness* has become a point of confusion, since there apparently is a large discrepancy in the literature on its formal definition. Despite this, the literature either on numerical methods or specifically on chemical kinetics often adopts a particular definition of stiffness, which may be better categorized as either a measure of stiffness or an underlying cause.

In addition to finding various "definitions" of stiffness, it is common to describe stiffness as a property of a particular instance of some problem. However, [15] and [19] more accurately describe stiffness as an observable phenomena, rather than a property of differential equations. This is more appropriate since we may construct examples that exhibit stiffness when a particular method is applied, but do not when another is used.

We aim to explore the nebulous nature of stiffness, showing commonalities and discrepancies between the commonly found explanations; unlike most explanations we maintain a distinction between metrics and causes of stiffness. In our numerical experiments we show how stiffness is able to influence the error propagation through the numerical integration. Additionally, we attempt to encapsulate these various notions of stiffness and develop an appropriate definition, which is largely taken from [19] and [25].

Literature on stiffness is in moderate agreement when stating that the most early attempts at defining stiffness are derived from a simple observation concerning the application of numerical methods. The following statements are commonly taken in the literature as definitions of stiffness.

Statement 1. *Differential equations with initial values are said to be stiff if the application of implicit methods perform exceedingly better than of explicit ones.*

Though [15] adopts this as its primary definition of stiffness, [25] addresses that there is not adequate as a mathematical framework for the theoretical

analysis of numerical methods. Consequently, this definition is insufficient in providing mathematical insight, though it is properly motivated and validated by what is seen in numerical experiments.

The most commonly found attempt at defining stiffness concerns the stepsize of the numerical method being used.

Statement 2. *Stiffness occurs when stability requirements, rather than accuracy, constrain the stepsize.*

We have not explored the concept of numerical stability, but let suffice to say that the *stability* of a numerical method is concerned with the accumulation of error, whereas accuracy refers to the error present at any particular point in the integration process. For some numerical methods, [19] further demonstrates that the error at the first step may be significantly higher for well known stiff problems over others. Therefore, separating stability from accuracy is not as apparent as Statement 2 may imply.

Statement 2 accurately suggests that stiffness is related to the application of a numerical method to a particular differential equation, and not a property of either one individually. Nonetheless, Statement 2 suffers from the same deficiency as Statement 1, namely that there is little mathematical analysis to be done for either explanatory or management purposes.

Some less commonly found statements about stiffness include arguments about the components of the numerical vector solution having large numerical disparity or is approximately exponential on some small initial interval. Again, these are no less vague that either Statement 1 or 2.

Literature sometimes states that stiffness is the event, occurrence, or observation that one or more of some criteria are satisfied. Viewing such attempts at defining stiffness by some Boolean condition is aligned with mathematical frameworks for making proper definitions. However, the criteria found in literature on stiffness give metrics or detection mechanisms, not some specific condition that must be satisfied. Therefore, the following statements about stiffness we consider to be detection mechanisms, rather than defining qualities. The origin of the most commonly found mechanism lies in noting that when a differential equation of the form given in Equation

19 is expanded by Taylor's Theorem to the include its first derivative, we have

$$\dot{x}(t_0 + h) = f(t_0, x_0) + \left(\frac{\partial}{\partial t} f(t_0, x_0) + \mathbf{D}f(t_0, x_0) \cdot \dot{x}(t_0) \right) h + O(h^2) \quad (33)$$

where $\mathbf{D}f$ is the Jacobian matrix. For the solution curve $x(t)$, we have a Taylor expansion

$$x(t_0 + h) = x_0 + \dot{x}(t_0)h + O(h^2) \quad (34)$$

Solving for $\dot{x}(t_0)h$ in Equation 34, substituting into Equation 33, and omitting the $O(h^2)$ term, we derive the equation

$$\dot{x}(t_0 + h) = f(t_0, x_0) + \frac{\partial}{\partial t} f(t_0, x_0)h + \mathbf{D}f(t_0, x_0) \cdot (x(t_0 + h) - x_0) \quad (35)$$

There are two interpretations to Equation 35, as [30] explains: one as a *quadrature*, and a second as a linear differential equation, both involving polynomials in h; the latter is preferred for large systems and we follow [30] and write Equation 35 as

$$\dot{x}(t_0 + h) = \mathbf{D}f(t_0, x_0) \cdot (x(t_0 + h) - x_0) + p(h) \quad (36)$$

where the components of p are polynomials in h. Let $\mathbf{A} = \mathbf{D}f(t_0, x_0)$ having dimension $n \times n$; the exact solution to 36 is

$$x(t_0 + h) = e^{\mathbf{A}h} x_0 + e^{\mathbf{A}h} \int_0^h e^{-\mathbf{A}h} p(\tau) d\tau \quad (37)$$

using matrix exponential notation, [30]. The term $e^{\mathbf{A}t}$ in Equation 37 is often written as $\sum_{i=1}^{n} \kappa e^{\lambda_i t} v_i$, where $v_i \in \mathbb{C}^n$ is the corresponding eigenvector to λ_i. We choose Definition 13 as our definition of the matrix exponential under the assumption that each eigenvalue $\lambda_i \in \mathbb{C}$ is distinct. This assumption is critical, since the eigenvalues play an fundamental role in studying the nature of $e^{\mathbf{A}t}$.

Definition 20. *From Equation 37, the transient of $x(t)$ is the term $\sum_{i=1}^{n} e^{\mathbf{A}t}$ and the steady-state component of $x(t)$ is the term*

$$e^{\mathbf{A}h} \int_0^h e^{-\mathbf{A}h} p(\tau) d\tau$$

Suppose $\mathbf{Re}(\lambda_i) < 0$; then $\lim_{t\to\infty} e^{\mathbf{A}t} = 0$, which implies that the transient disappears and $x(t)$ approaches steady state asymptotically as $t \to \infty$, [21]. If $\lambda_i \in \mathbb{R}$, then $e^{\mathbf{A}t}$ decays monotonically and sinusoidally otherwise.

For large $|\mathbf{Re}(\lambda_i)|$, the corresponding transient has a faster rate of decay and conversely for small $|\mathbf{Re}(\lambda_i)|$; we conventionally refer to these scenarios as "fast" transients and "slow" transients. Since the eigenvalues are presumed unique, we may discuss the magnitude of their effect on the transient by constructing the sequence

$$|\mathbf{Re}(\lambda_1)| \geq |\mathbf{Re}(\lambda_2)| \geq \cdots \geq |\mathbf{Re}(\lambda_n)|$$

from which it is clear that the transient involving λ_1 is the fastest and that involving λ_n is the slowest.

Consideration of ordering the eigenvalues leads to the most commonly found statement on stiffness by giving metrics.

Statement 3. *If the Jacobian matrix of a differential system has unique eigenvalues in the open left plane, then stiffness is the phenomena of the ratio*

$$\frac{|\lambda_1|}{|\lambda_n|} > 0$$

being large.

This is perhaps the most commonly found attempt at properly analyzing stiffness by metrics, as it is found in [3], [4], and [30]. Clearly, by truncating Equation 37 to not include the steady state, an initially exponential approximation is obtained; this appears to validate the observation that stiffness is when the numerical solution is nearly exponential on some small interval. Both [15] and [19] assert Statement 3 as being sound, but also that it is not fully inclusive, as is shown by an example in [19], though the example is not in the context of chemical reaction networks.

Statement 4. *Stiffness occurs when some components of the solution decay rather quickly compared to others.*

The eigenvalues of the Jacobian are $\lambda_1 = 0, \lambda_2 = -k_2, \lambda_3 = -k_1$. The inadequacy of defining stiffness by Statement 3 is best demonstrated by considering the limiting case when $k_1 = k_2$; we observe that there is a noticeable

performance difference for small k_1, k_2 compared to large k_1, k_2, despite the stiffness ratio remaining 1. Though, Statement 3 is somewhat inadequate, it is the foundation for most other measures of stiffness.

Statement 5. *Stiffness occurs when a system is "locally exponential."*

Statement 5 is simply an interpretation of the transient initially dominating the steady state, usually by truncation to include only the first term of the Taylor series. While Statement 5 is observed in many situations, a system is locally exponential if the stiffness ratio is large. In other words, it is not sufficient to simply have a large eigenvalue, but rather a large difference between the largest and smallest is needed. Conversely, if a system has a large stiffness ratio, then it will initially be exponential.

Each of the discussed statements concerning stiffness are rooted in what is commonly seen. Consequently, it is possible to construct counter examples and thereby leaving stiffness as a vague concept. We propose a definition having a genesis in [19], which attempts to include all of the previously discussed ideas.

Definition 21. *Stiffness is the event that a numerical method with a finite region of absolute stability, when applied to a system with any initial conditions, is forced to use a stepsize which is excessively small in relation to the smoothness of the exact solution in a certain interval of integration.*

Unlike most claims about stiffness in the literature, our definition stresses that a system is not independently stiff, but may be considered such when a particular method is applied. What we mean by an "excessively small" stepsize depends on the stage of integration; since on some time interval the system or its solution may be more smooth than elsewhere, a small stepsize would be considered excessive on that region.

Despite being inclusive of other definitions of stiffness and being well formulated, our definition has an inherent flaw. Specifically, the exact solution is quite rarely known, making comparison to numerical approximations nearly impossible; in fact, if the exact solution is known, then numerical approximation are of low utility. For such reasons, [25] is dissatisfied with complex definitions of stiffness and reverts to the basic statement: stiffness is the even

that an excessive reduction in the stepsize is needed to achieve minimal accuracy. Indeed, this is simply what the mathematician will experience when analyzing a differential equation by some numerical method.

Chemical Network

$$\mathrm{A} \xrightarrow{k} \mathrm{B}$$
$$2\,\mathrm{A} \xrightarrow{k} \mathrm{B}$$
$$3\,\mathrm{A} \xrightarrow{k} \mathrm{B}$$
$$\mathrm{A} + \mathrm{B} \xrightarrow{k} \mathrm{C}$$
$$\mathrm{A} + 2\,\mathrm{B} \xrightarrow{k} \mathrm{C}$$
$$\mathrm{A} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathrm{B}$$
$$\mathrm{A} \xrightarrow{k_1} \mathrm{B} \xrightarrow{k_2} \mathrm{C}$$
$$\mathrm{A}_i \xrightarrow{k_i} \mathrm{B}_i$$
$$2\mathrm{A}_i \xrightarrow{k_i} \mathrm{B}_i$$

Figure 9: Nonstiff chemical networks taken from [3]

5 Numerical Experiments

5.1 Nonstiff Numerical Experiments

In [3] and [4], Taylor's method is claimed to perform well for a certain class of models with specified rate constants and initial values, which are listed in Figure 9. These models feature reactions that are either nonstiff or exhibit a mild degree of stiffness.

For each of the models in Figure 9 it is possible to solve the system by standard or previously discussed techniques. The following examples give the solutions for each of these models without regard to any specific rate constants or initial values.

Example 5.1. The species formation function for the chemical reaction network $\mathrm{A} \xrightarrow{k} \mathrm{B}$ is often studied in the literature on numerical solutions to differential equations, independently of any chemical network context. An examples of this reaction is the unimolecular first order reaction $\mathrm{CH_3NC} \xrightarrow{k} \mathrm{CH_3CN}$. The exact solution for the time evolution of A is well known, and is the scalar

version of the differential system presented in Theorem 3, which is

$$x = x_0 e^{-kt}$$

.

Example 5.2. Consider the reaction $2\,\mathrm{A} \xrightarrow{k} \mathrm{B}$. An example of such a reaction is the recombination of two identical radicals, such as two methyl radicals: $2\,\mathrm{CH_3} \xrightarrow{k} \mathrm{C_2H_6}$. Since we are solving the equation

$$f(x(t)) := \frac{dx}{dt} = -2kx^2$$

we may rearange the differentials to obtain

$$\int_{x_0}^{x_t} \frac{dx}{x^2} = -2k \int_0^t dt$$

which results in

$$\frac{1}{x} = \frac{1}{x_0} + 2kt$$

Example 5.3. Using similar techniques to the previous reaction, we may solve the species formation function for the reaction $3\,\mathrm{A} \xrightarrow{k} \mathrm{B}$ to obtain

$$\frac{1}{2}\left(\frac{1}{x^2} - \frac{1}{x_0^2}\right) = 3kt$$

Example 5.4. To integrate the species formation function for the reaction $\mathrm{A} + \mathrm{B} \xrightarrow{k} \mathrm{C}$, define the *progress variable* $z := (x_1)_0 - (x_1)_t = (x_2)_0 - (x_2)_t$, under the notation of Figure 9. The the species formation function, as given in Figure 9 may be reformulated as

$$\frac{dz}{dt} = k\left((x_1)_0 - (x_1)_t\right)\left((x_2)_0 - (x_2)_t\right)$$

and we integrate by

$$\int_{z(0)}^{z(t)} \frac{dz}{\left((x_1)_0 - z\right)\left((x_2)_0 - z\right)} = k \int_0^t dt \tag{38}$$

Temporarily, let

$$\varkappa := \int \frac{dz}{\left((x_1)_0 - z\right)\left((x_2)_0 - z\right)}$$

After separating the variables, we use the method of partial fractions to obtain

$$\varkappa = \int \frac{dz}{((x_1)_0 - (x_2)_0)\,((x_2)_0 - z)} - \int \frac{dz}{((x_1)_0 - (x_2)_0)\,((x_1)_0 - z)} \tag{39}$$

We solve the right side of Equation 39 and equate to the left of Equation 40. The solution to the species formation function is therefore

$$\frac{1}{(x_1)_0 - (x_2)_0} \ln\left(\frac{(x_2)_0 \cdot (x_1)_t}{(x_1)_0 \cdot (x_2)_t}\right) = kt \tag{40}$$

We note that data is able to be plotted in the form of the left side of Equation 41 against t; or for any particular value of t, a trajectory plot of the two concentrations may be plotted.

Example 5.5. Solving the system for the reaction $\mathrm{A} + 2\,\mathrm{B} \xrightarrow{k} \mathrm{C}$ is done in [26] by supposing that the concentration of A is significantly greater than that of species B, such that the concentration of A effectively does not change during the reaction time. Defining $x := [A]$, we reformulate the species formation function as

$$\dot{x} = -2k'x^2$$

meaning that we have "reduced" the original third order reaction to a pseudo second order reaction where the rate coefficient depends on the concentration of B.

If the reaction occurs such that the initial concentrations of the dissimilar reactants A and B are equal, then by defining the progress variable z such that $(x_2)_t = (x_2)_0 - 2z$ and $(x_1)_t = (x_1)_0 - z$ we may reformulate the species formation function as

$$\frac{dz}{dt} := k\,((x_1)_0 - z)\,((x_2)_0 - 2z)^2$$

which is separable by

$$\frac{dz}{((x_1)_0 - z)\,((x_2)_0 - 2z)^2} = kdt$$

Let $\varkappa := (x_2)_0 - 2(x_1)_0$. We may use the method of partial fractions to derive

$$\frac{1}{\varkappa}\left(\frac{1}{(x_2)_0} - \frac{1}{(x_2)_t}\right) + \frac{1}{\varkappa^2} \ln\left(\frac{(x_2)_t \cdot (x_1)_0}{(x_2)_0 \cdot (x_1)_t}\right) = kt$$

Instances of chemical reactions satisfying this form, and consequently having this species formation function are well known. The canonical example is the formation of water $2\,H_2 + O_2 \xrightarrow{k} 2\,H_2O$; others include the gas phase reaction between nitric oxide and oxygen: $2\,NO + O_2 \xrightarrow{k} 2\,NO_2$.

Example 5.6. The reversible reaction $A \underset{k_2}{\overset{k_1}{\rightleftharpoons}} B$ is solved using the matrix exponential in Example 3.1.

Example 5.7. The reactions $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ constitute a system of first order "consecutive" reactions. We have already solved the system $A \xrightarrow{k_1} B$, and from Example 5.1 has the solution

$$x_1 = (x_1)_0 e^{-k_1 t}$$

Substituting into the modeling function for the evolution of B, we have

$$\frac{dx_2}{dt} + k_2 x_2 = k_1 (x_1)_0 \cdot e^{-k_2 t}$$

which is solved by standard methods. The time evolution of species B is therefore given by

$$x_2 = (x_2)_0 \cdot e^{-k_2 t} + \frac{k_1 (x_1)_0}{k_2 - k_1} \left(e^{-k_1 t} - e^{-k_2 t} \right)$$

To describe the time evolution of species C, must assume the conservation of mass: $(x_1)_0 = x_1 + x_2 + x_3$. By substituting in the equations describing the time dependence of the concentrations for both A and B, we have

$$x_3 = (x_1)_0 - \left((x_1)_0 \cdot e^{-k_1 t} \right) - \left((x_2)_0 \cdot e^{-k_2 t} + \frac{k_1 (x_1)_0}{k_2 - k_1} \left(e^{-k_1 t} - e^{-k_2 t} \right) \right)$$

from which we may compactly write

$$x_3 = (x_2)_0 \cdot e^{-k_2 t} + (x_1)_0 \cdot \left(1 - \frac{k_2}{k_2 - k_1} e^{-k_1 t} - \frac{k_1}{k_2 - k_1} e^{-k_2 t} \right)$$

Example 5.8. The set of reactions having components $A_i \xrightarrow{k_i} B_i$ is a vectorized version of the reaction $A \xrightarrow{k_i} B$, which was solved in Example 5.1. The solution to the time evolution of species A_i is therefore $x_i = (x_i)_0 \cdot e^{-k_i t}$, which may be vectorized to yield the solution to the vectorized species formation function.

Example 5.9. A vectorized version of the system $2\,\mathrm{A} \xrightarrow{k} \mathrm{B}$, which we previously solved, is precisely the system having components $2\mathrm{A}_i \xrightarrow{k_i} \mathrm{B}_i$. To solve this system, we simply vectorize the solution to $2\,\mathrm{A} \xrightarrow{k} \mathrm{B}$.

Numerical solutions for each of the systems in Figure 9 is of utility when compared to the actual solution by providing comparative analysis. However, insight into which numerical methods perform better for instances of the species formation function is best done when the exact solution is not known, and therefore investigating properties of the numerical procedures or consideration of the context of their applications are the only options.

5.2 Implementation of `TR-BDF2` in `ode23tb`

Matlab®'s `ode23tb` is an implementation of `TR-BDF2`, an implicit Runge-Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a backward differentiation formula of order two. The major components of the implementation of the `BDF2` portion of the algorithm were developed in the context of device simulation with the goal of minimizing memory usage at a time when storage space was costly; because of its success in other contexts, `TR-BDF2` has been scaled and optimized for commercial use, only one of which is `ode23tb`.

As an implicit Runge-Kutta formula, `TR-BDF2` is quite unusual, since it involves an explicit stage formula. Consequently, most of the canonical and herein described theorems do not apply in a global sense, though assumptions leading to better performance, such as from Theorem 11 are inferred from [16], professing that certain desirable properties that are preserved. The `BDF2` algorithm and its associated error estimate may be viewed as an implicit Runge-Kutta pair of orders 2 and 3, specifically a diagonally implicit Runge-Kutta with Butcher tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \varsigma & \eta & \eta & 0 \\ 1 & \jmath & \jmath & \eta \\ \hline & \jmath & \jmath & \eta \\ \hline & (1-\jmath)/3 & (3\jmath+1)/3 & \eta/3 \end{array}$$

where $\varsigma = 2 - \sqrt{2}$, $\jmath = \varsigma/2$, and $\eta = \sqrt{2}/4$. The vector

$$\hat{b}^T := \left[\frac{1-\jmath}{3}, \frac{3\jmath+1}{3}, \frac{\eta}{3}\right]$$

has parameters which meet proprietary conditions for possible stepsize control.

We note that `TR-BDF2` is "almost" an singly diagonally implicit Runge-Kutta method; since the first stage is explicit, there is no nonlinear equation to be solved for its evaluation. Moreover, the first stage of a step is the same as the last stage from the end of the previous step. Because of the following algorithm properties, [16] advocates the use of `TR-BDF2`:

- `TR-BDF2` is "first-same-as-last" and hence there are only two implicit stages to evaluate per step, rather than three.
- `TR-BDF2` provides a "free" asymptotically correct error estimate.
- `TR-BDF2` possesses "desirable" stability properties.
- All the stages are evaluated within the step interval.

Associated with its Butcher tableau, `TR-BDF2` must satisfy $c_2 = 2\eta$ and the order conditions up to order two and the companion formula, the order conditions up to order three:

$$\begin{array}{l} b_1 + b_2 + b_3 = 1 \\ b_2 c_2 + b_3 = \frac{1}{2} \\ \hat{b}_1 + \hat{b}_2 + \hat{b}_3 = 1 \\ \hat{b}_2 c_2 + \hat{b}_3 = \frac{1}{2} \\ \hat{b}_2 c_2^2 + \hat{b}_3 = \frac{1}{3} \\ \hat{b}_2 c_2 \left(\frac{1}{2} c_2 - b_3\right) + \hat{b}_3 \left(\frac{1}{2} - b_2 c_2 - b_3\right) = 0 \end{array} \tag{41}$$

Under these conditions, [16] reports that `TR-BDF2` is extremely performant under high degrees of stiffness. At the initialization of the algorithm, or at a restart, the stage must be formed by the explicit stage, and it can be formed in this way at any step; subsequent steps are obtained by rescaling the current approximate solution as the last stage in the previous step.

As in [16], `TR-BDF2` offers an accurate alternative when other conventional numerical methods fail; its implementation in `ode23tb` is suggested as the alternative when default Matlab® methods, such as `ode15s` and `ode45` do not perform as desired.

5.3 Stiff Numerical Experiments

As a demonstration of the failure of certain methods in the presence of stiffness, [16] and [15] consider the system of differential equations we examine as Example 2.3. The mass action network for Example 2.3 is given by

$$\begin{aligned} \mathrm{A} &\xrightarrow{k_1} \mathrm{B} \\ 2\,\mathrm{B} &\xrightarrow{k_2} \mathrm{B} + \mathrm{C} \\ \mathrm{B} + \mathrm{C} &\xrightarrow{k_3} \mathrm{A} + \mathrm{C} \end{aligned}$$

from which we have the system of equations

$$\begin{array}{ll} \dot{x}_1 = k_1 x_1 + k_3 x_2 x_3 & x_1 := [A] \\ \dot{x}_2 = k_1 x_1 - k_3 x_3 - k_2 x_2^2 & x_2 := [B] \\ \dot{x}_3 = k_2 x_2^2 & x_3 := [C] \end{array} \tag{42}$$

Example 5.10. Consider the above mass action system, having species formation function given by 43. If we take $k_1 = k_2 = k_3 = 1$, then according to our definition of stiffness, as given in Definition 21, Equation 43 is not stiff. We conclude this by observing that the stepsize is not overly constrained in `ode45`, `ode23tb`, or `taylor4th` for any initial values or any choice of rate constants.

The numerical output for applying `ode45`, `ode23tb`, or `taylor4th` to this example are given in Figures 10, 11, and 12, respectfully. The execution of each algorithm was done with a constant step of 0.01 and initial condition $\vec{x}_0 = [1, 0, 0]^T$.

Assuming the conservation of concentrations, $\sum_i (x_i)_0 = \sum_i (x_i)_t$ must hold at any time t on the interval of integration. Since $\sum_i (x_i)_0 = 1$ for this chemical network, each algorithm must satisfy $1 - \sum_i (x_i)_t = 0$. Figures 13, 14, and 15 chart the progression of error under the conservation of mass for

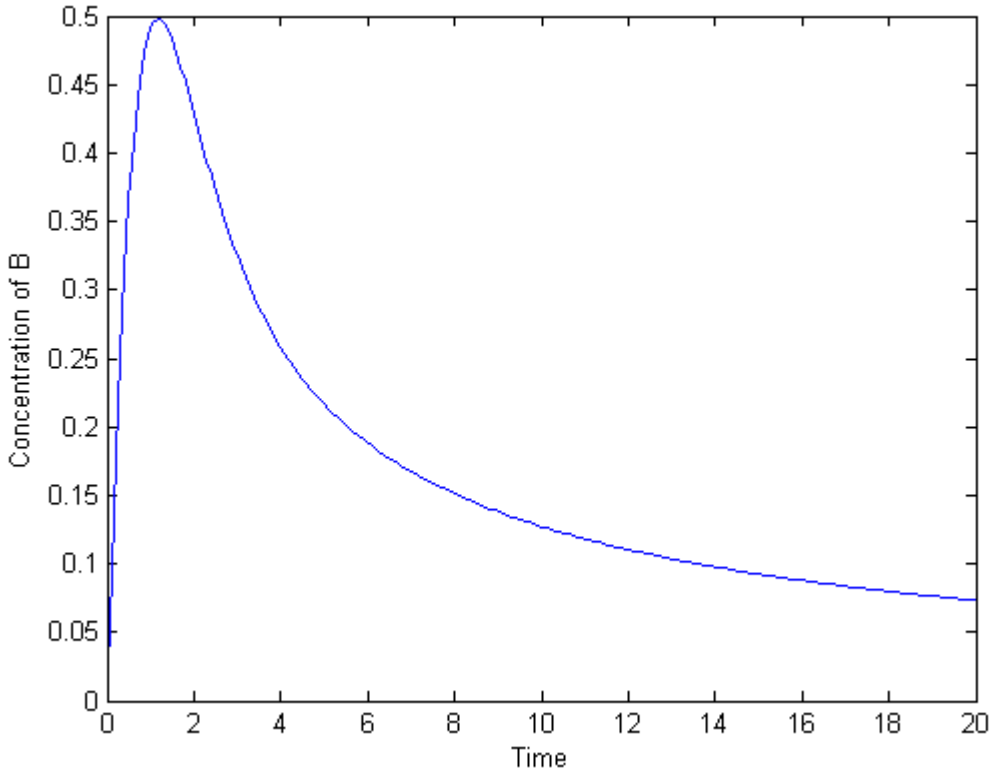


Figure 10: `ode45` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = k_2 = k_3 = 1$

`ode45`, `ode23tb`, or `taylor4th`, respectively.

From the smoothness of the output for `ode45`, `ode23tb`, and `taylor4th` and the small bounds on the error for each, we consider the system of differential equation successfully numerically integrated.

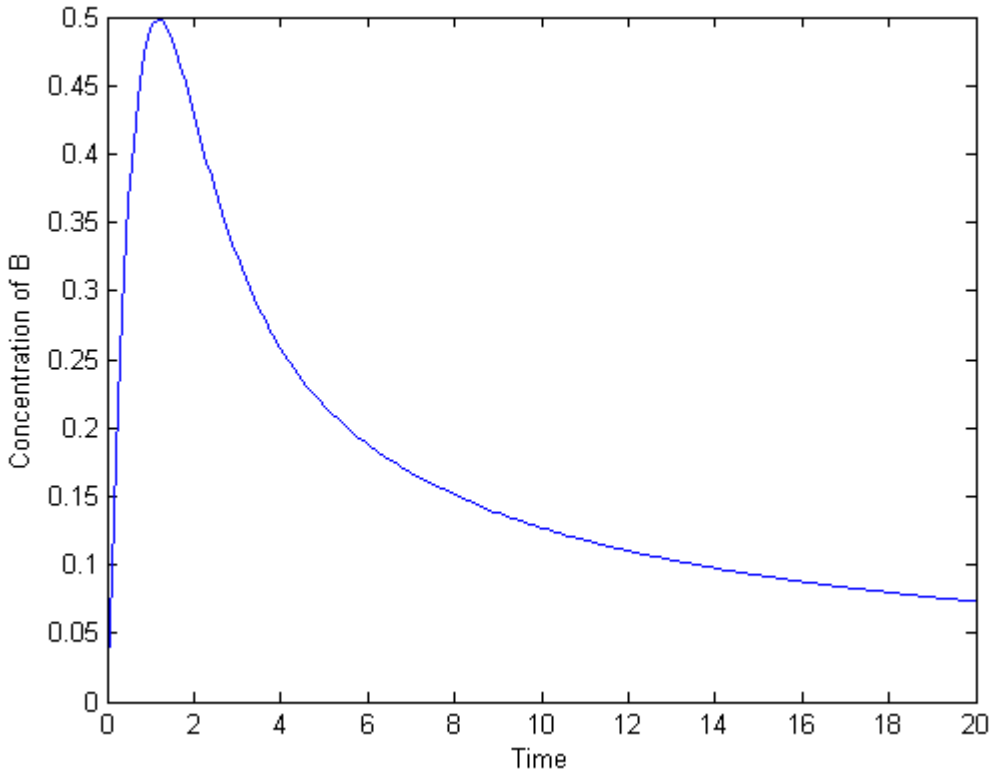


Figure 11: `ode23tb` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = k_2 = k_3 = 1$

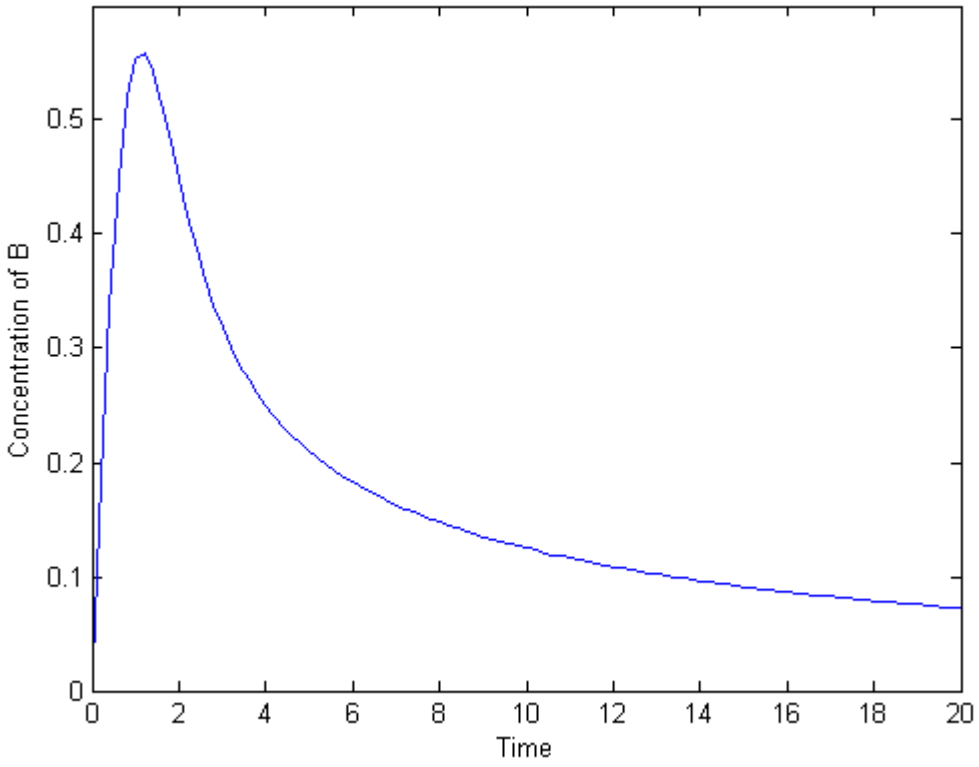


Figure 12: `taylor4th` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = k_2 = k_3 = 1$

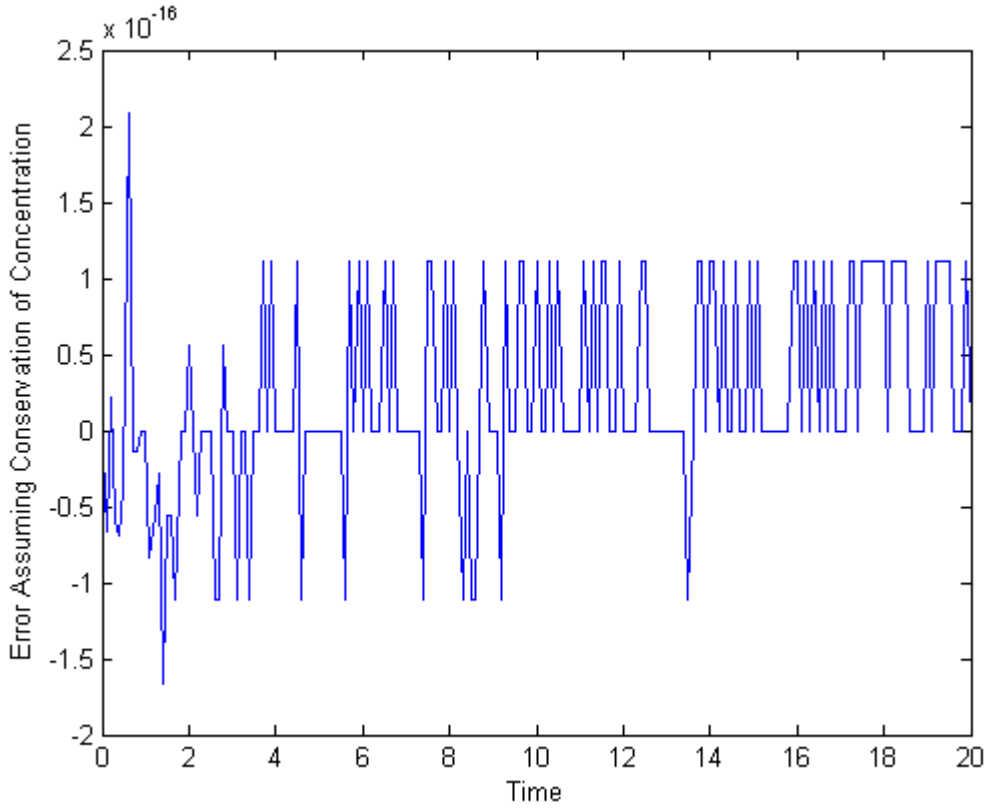


Figure 13: Error progression for `ode45` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = k_2 = k_3 = 1$ assuming conservation of concentration

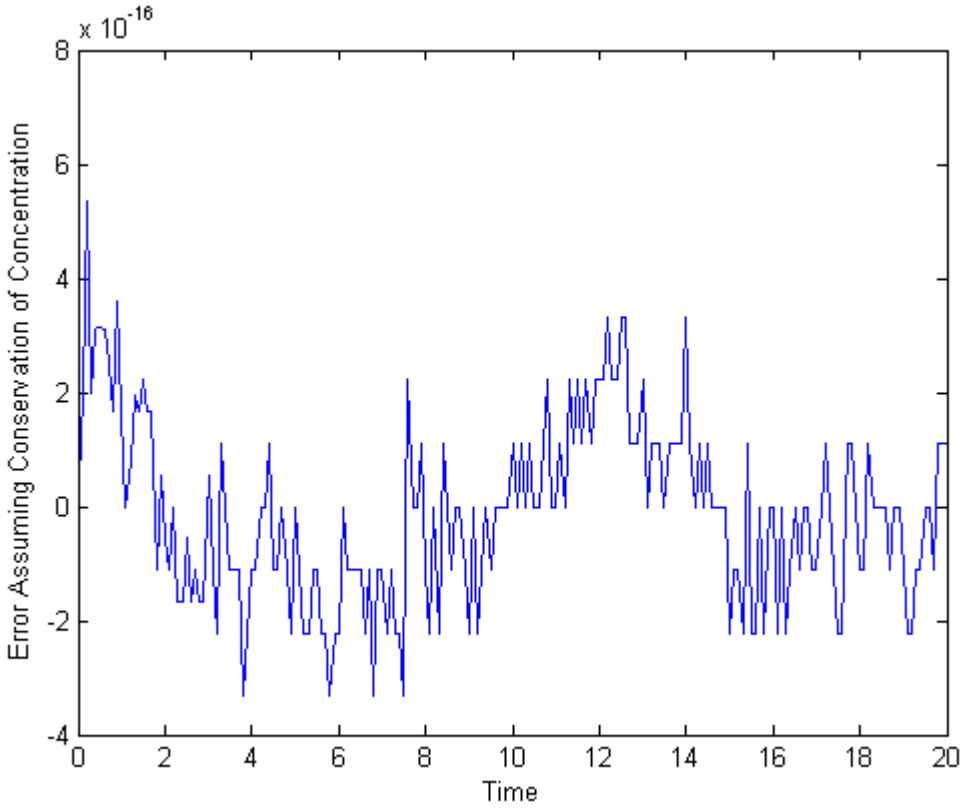


Figure 14: Error progression for `ode23tb` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = k_2 = k_3 = 1$ assuming conservation of concentration

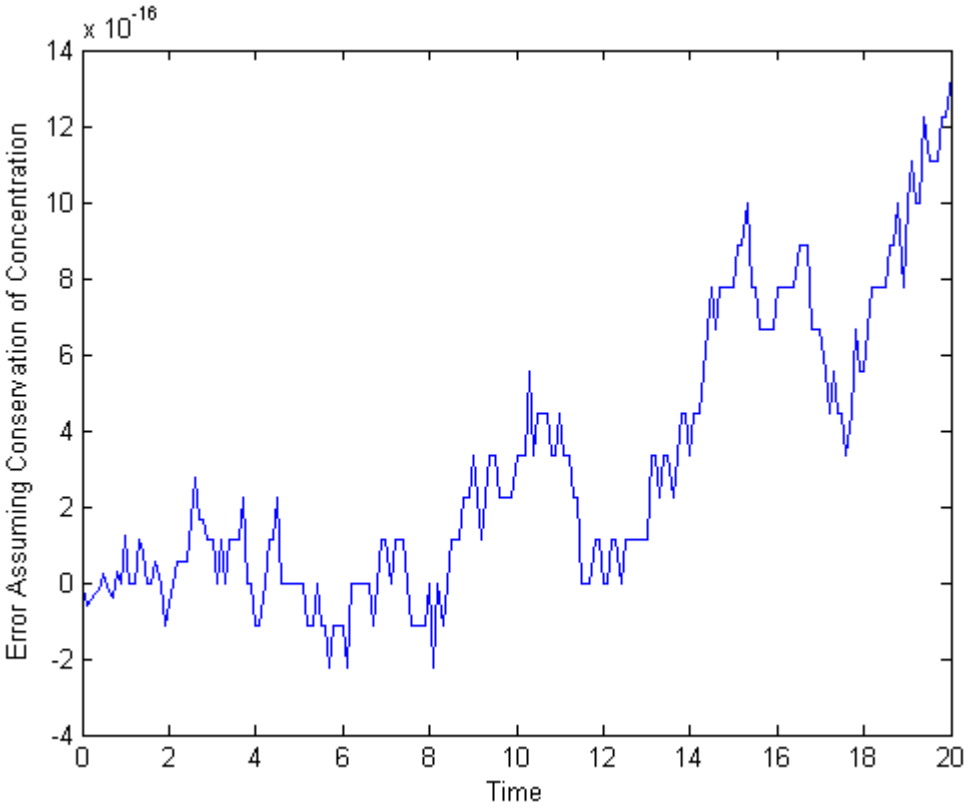


Figure 15: Error progression for `taylor4th` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = k_2 = k_3 = 1$ assuming conservation of concentration

In addition to the conservation of concentrations, we could additionally consider how closely each numerical routine follows the system's moiety constraints. However, for this particular example, the stoichiometric matrix is given by

$$\mathcal{S} := \begin{bmatrix} \text{-1} & 0 & 1 \\ 1 & \text{-1} & \text{-1} \\ 0 & 1 & 0 \end{bmatrix} \tag{43}$$

and therefore has augmented matrix

$$\mathbf{W} := \left[\begin{array}{ccc|ccc} \text{-1} & 0 & 1 & 1 & 0 & 0 \\ 1 & \text{-1} & \text{-1} & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \tag{44}$$

which has $\text{rank}(\mathbf{W}) = 3$. By Theorem 1, we have $3 - \text{rank}(\mathbf{W}) = 0$, which indicates that there are no moiety constraints present.

Example 5.11. The supposition that $k_1 = k_2 = k_3 = 1$ as in Equation 43 is seldom encountered in practice and therefore does not serve as a representative of the science; moreover, [30] claims that large reaction rates increase the stiffness of the system; our observations are consistent with this supposition. To demonstrate when this system exhibits stiffness, suppose $k_1 = 0.04$, $k_2 = 3 \cdot 10^7$, and $k_3 = 1 \cdot 10^4$, which is likewise considered in [15] and [16].

The numerical output for applying `ode45`, `ode23tb`, or `taylor4th` to 43 when $k_1 = 0.04$, $k_2 = 3 \cdot 10^7$, and $k_3 = 1 \cdot 10^4$ are given in Figures 17 and 18, respectfully. The execution of each algorithm was again done with a constant step of 0.01 and initial condition $\vec{x}_0 = [1, 0, 0]^T$.

Notice that `ode45` struggles with the numerical integration, which is observable by the jagged behavior. This is also reflected in Figure 19, which tracks the errors under the assumption that the conservation of mass holds: $1 - \sum_i (x_i)_t = 0$. Conversely, `ode23tb` integrates with more success, as seen in the smoothness of Figure 18 and by the error tracking in Figure 20.

`taylor4th` after only a few iterations fails to converge. We observe this when the reaction rates are changed, but remain stiff. Also, under the rates $k_1 = 0.04$, $k_2 = 3 \cdot 10^7$, and $k_3 = 1 \cdot 10^4$, alteration of the initial conditions eventually produces the same result. We provide the numerical output for `taylor4th` assuming these rates and the initial condition $\vec{x}_0 = [1, 0, 0]^T$.

t	0	0.1	0.2	0.3	0.4	...	20
x_1	1	0.99335	-1.9377e+020	-1.9373e+157	NaN	...	NaN
x_2	0	15.959	-2.7928e+028	-4.5849e+164	NaN	...	NaN
x_3	0	-15.952	-2.7928e+028	-4.5849e+164	NaN	...	NaN

Figure 16: Cumulative error for `ode45`, `ode23tb`, and `taylor4th` applied to Example 5.10

We conclude that `taylor4th` is in fact the worst choice of the three algorithms; the commercial implicit Runge-Kutta solver, `ode23tb`, is the "best" of the three algorithms, which is consistent with the statements in [15] regarding implicit methods performing better than explicit ones. Because `taylor4th` is not a commercial product, unlike `ode23tb`, it lacks internal optimization. This is a potential cause of poor performance for stiff models, though the numerical explosion is not likely caused by the implementation.

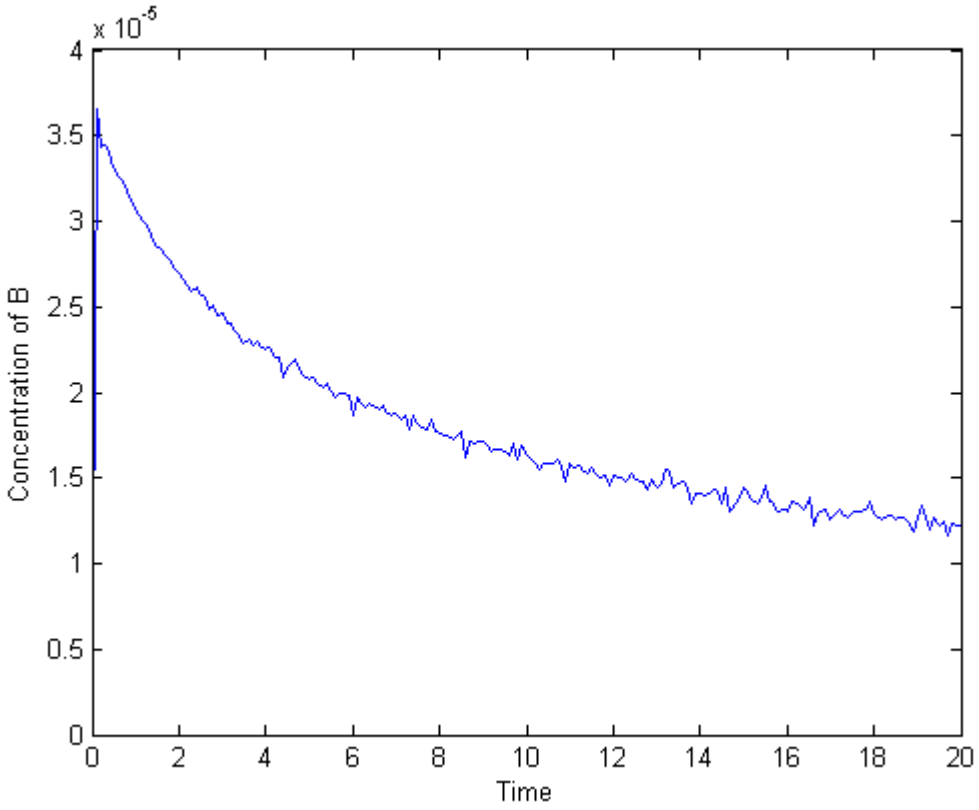


Figure 17: `ode45` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = 0.04$, $k_2 = 1 \cdot 10^4$, and $k_3 = 3 \cdot 10^7$

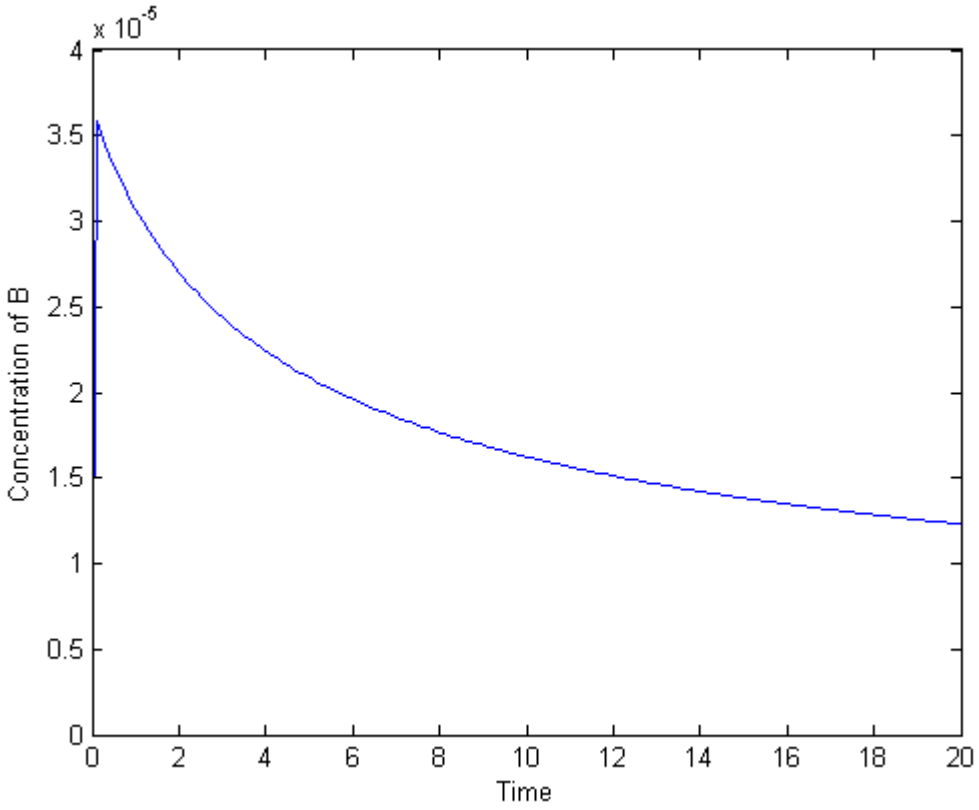


Figure 18: `ode23tb` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = 0.04$, $k_2 = 1 \cdot 10^4$, and $k_3 = 3 \cdot 10^7$

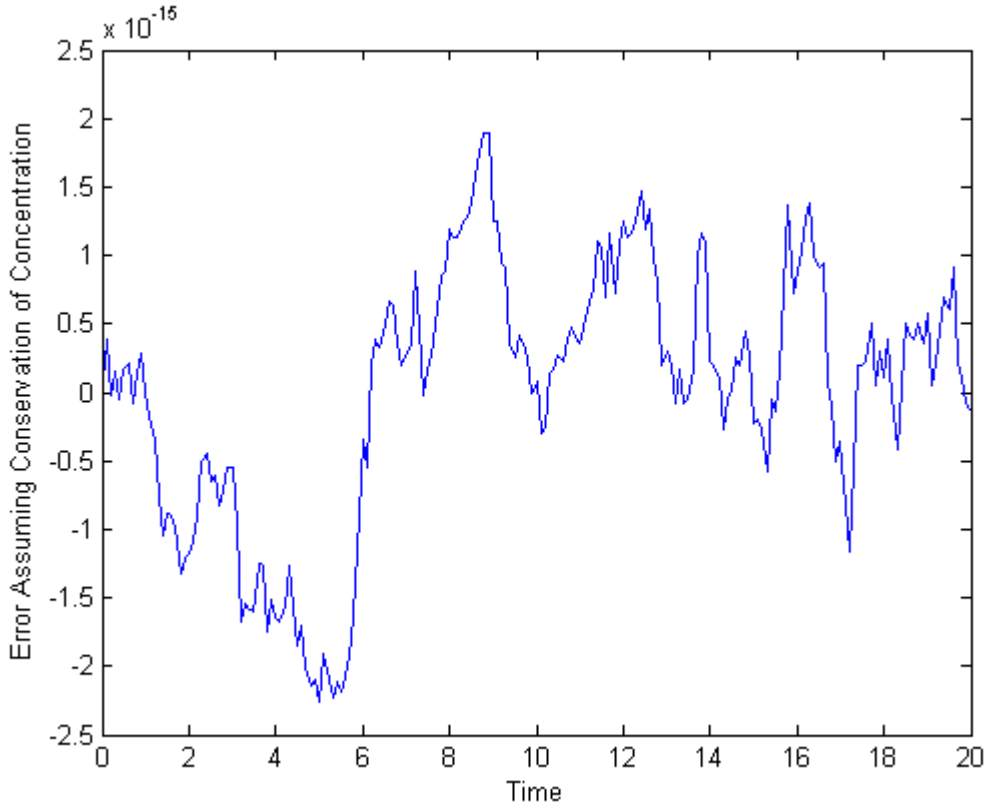


Figure 19: Error progression for `ode45` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = 0.04$, $k_2 = 1 \cdot 10^4$ assuming conservation of concentration

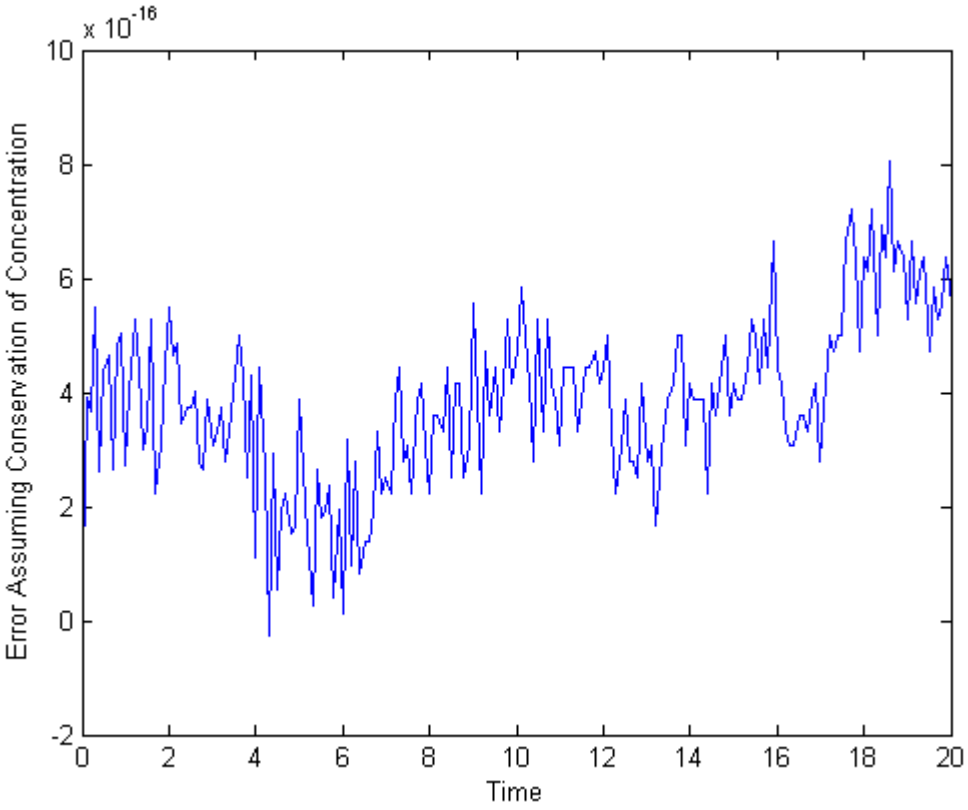


Figure 20: Error progression for `ode4ode23tb` applied to the species formation function given by the mass action network in Example 2.3, with reaction constants $k_1 = 0.04$, $k_2 = 1 \cdot 10^4$ assuming conservation of concentration

6 Conclusion

We herein consider the deterministic approach of modeling the time evolution of species' concentrations in a chemical system, where they are considered as a continuous, wholly predictable process which is governed by a set of differential equations, the species formation function. Our approach is motivated directly by the assumption that the chemical system follows the Law of Mass Action. The ability to numerically integrate instances of the species formation function is of scientific interest for those wanting to describe the concentration of species between the initial time t_0 and as time progresses to infinity.

Because applications of this effort include biopharmaceutical, accuracy in the numerical method is of great importance. As our experimentation demonstrates, instances of the species formation function exhibit stiffness, making the accuracy of output greatly dependant on the choice of numerical method. For the system given in Example 2.3 exhibiting stiffness, we cannot extend the hypothesis from [3] that the Taylor Series Method is comparably or better performing than other conventional methods. Moreover, for certain chemical networks we are able to use standard calculus techniques to gain a "closed form" equation describing the time evolution of species.

From the perspective of algorithm design, and in order to assess the quality of output from any choice of numerical method, we may suppose the Law of Conservation of Concentration holds for closed systems. Doing so enables recording the cumulative error of the method without consideration to a method's particular error properties. Additionally, we may initially detect moiety conservations to likewise give more conditions that the error accumulation should satisfy.

Due to the nebulous nature of stiffness and its influence on the quality of numerical results, further investigation on this subject is warranted. Such would continue to be of interest from the perspective of algorithm and software design, as well as for those concerned with the change of concentrations of species in any *in-vivo* or *in-vitro* system.

7 Future Directions

We herein consider the deterministic approach with regards to the time evolution of species' concentrations, where they are considered as a continuous, wholly predictable process which is governed by a set of differential equations, the species formation function. Alternatively, the stochastic approach regards the time evolution as a sort of random-walk process which is governed by a single differential-difference equation, referred to as the *master equation*. This modeling process is also found in the literature on chemical networks, as in [26]. Fairly simple arguments from kinetic theory show that the stochastic formulation of chemical kinetics has a firmer physical basis than the deterministic formulation, and therefore investigation on this approach may lend to more scientifically relevant results.

With regard to the literature on chemical networks by the deterministic approach, little research has been done concerning stiff equations. Analysis of stiff equations is mostly done in the general context of studying numerical methods, where many more conventional and unconventional methods have been developed for this exact purpose. Further analysis on stiff kinetic models may consist of examining the contextual causes of stiffness and various numerical methods that are able to address this specific concern. A potential advantage of focusing on chemical networks is by exploiting the polynomial form of the species formation function.

A unique analysis of the specific compositional properties of the species formation function is provided in [28]. Therein a series of necessary and sufficient conditions for what is called *lumping* and *expanding* is specified. Additionally, [28] addresses how lumping changes properties of the numerical solutions, which are either interesting from the point of view of the qualitative theory of differential equations or from the point of view of formal reaction kinetics. Investigation on this subject may lend to a better theory for mass action kinetics.

A Source Code for `taylor4th`

Herein we provide source code for the Matlab routine `taylor4th` which numerically approximates the solution to a first order differential equation by a fourth order Taylor Series Method. The calling sequences, input parameters, and output parameters are included in the code and described in Matlab comments. The input parameter `DIFF` is a string containing the name of an m-file defining the right-hand side of the differential equation. The routine is factorized such that the subsequent derivatives for the Taylor Method are computed prior to any iteration. Therefore, the symbolic derivatives serve simply as input to the evaluation m-file, which is named in `EVAL`. For clarity, the Taylor series is explicitly truncated with hard-coded factorial values and all variable names are appropriately named.

`taylor4th` is written such that it is applicable to either autonomous or non autonomous first order differential equations. The distinction between the two cases is specified in the defining m-file, `DIFF`. Also, the routine may be used for a single differential equation or a system, where either a vector or matrix of output values is returned. The factorization of the code is done in an attempt to maximize its applicability and extendibility.

```
function [wi, ti] = taylor4th ( DIFF, EVAL, t0, x0, tf, N )

%      Approximates the solution of the initial value problem
%
%                  x'(t) = DIFF( x(t) ),    x(t0) = x0
%
%      using the fourth-order Taylor method - this routine
%      will work for a system of first-order equations as
%      well as for a single equation.
%
%  calling sequences:
%      [wi, ti] = taylor4th ( DIFF, EVAL, t0, x0, tf, N )
%      taylor4th ( DIFF, EVAL, t0, x0, tf, N )
%
%  inputs:
```

```
%      DIFF     string containing name of m-file defining the
%                 right-hand side of the differential equation
%                 and its first three symbolic derivatives with
%                 respect to the independent variable. Depends
%                 on the Symbolic Math Toolbox. The prototype
%                 for the m-file should be:
%                 [d1f, d2f, d3f, d4f] = DIFF( )
%                 where f is the value of the right-hand side
%                 function, d1f is the value of the derivative
%                 with respect to the independent variable,
%                 d2f is the value of the second derivative
%                 and d3f is the value of the third derivative
%
%       EVAL   string containing name of m-file defining the
%                 substitution of the symbolic variables in the
%                 right-hand side of the differential equation
%                 with the independent variable values.  Depends
%                 on the Symbolic Math Toolbox. The prototype
%                 for the m-file should be:
%                 [f ft ft2 ft3] = EVAL( d1f, d2f, d3f, d4f, x0 )
%
%       t0        initial value of the independent variable
%
%       x0        initial value of the dependent variable(s)
%                 if solving a system of equations, this should
%                 be a row vector containing all initial values
%
%       tf        final value of the independent variable
%
%       N         amount of uniformly sized time steps taken to
%                 advance the solution from t = t0 to t = tf
%
%  output:
%       wi        vector / matrix containing values of the
%                 numerical solution to the differential equation
%
%       ti        vector containing the values of the independent
%                 variable at which an approximate solution has
```

```
%                 been obtained
%

neqn = length ( x0 );
ti = linspace ( t0, tf, N+1 );
wi = [ zeros( neqn, N+1 ) ];
wi(1:neqn, 1) = x0';

h = ( tf - t0 ) / N;

[d1f, d2f, d3f, d4f] = feval ( DIFF);

for i = 1:N
    [f ft ft2 ft3] = feval ( EVAL, [d1f, d2f, d3f, d4f], x0 );

    x0 = x0 + (h * f) + (h^2 * ft / 2) +
     (h^3 * ft2 / 6) + (h^4 * ft3 / 24);
t0 = t0 + h;

    wi(1:neqn,i+1) = x0';
end;
```

B Example Implementation of Stiff Kinteic Models

```
function [d1f, d2f, d3f, d4f] = DIFF()

syms x y z real


k=0.04;
v=(3*10^7);
r=(1*10^4);


d1f(1) = -k*x+r*y*z;
d1f(2) = k*x-r*y*z-v*y^2;
d1f(3) = v*y^2;


temp(1) = diff(d1f(1),x) + diff(d1f(1),y) + diff(d1f(1),z);
temp(2) = diff(d1f(2),x) + diff(d1f(2),y) + diff(d1f(2),z);
temp(3) = diff(d1f(3),x) + diff(d1f(3),y) + diff(d1f(3),z);
d2f = transpose(temp)*d1f;


temp(1) = diff(d2f(1),x) + diff(d2f(1),y) + diff(d2f(1),z);
temp(2) = diff(d2f(2),x) + diff(d2f(2),y) + diff(d2f(2),z);
temp(3) = diff(d2f(3),x) + diff(d2f(3),y) + diff(d2f(3),z);
d3f = transpose(temp) * d2f;


d4f(1) = diff(d3f(1),x) + diff(d3f(1),y) + diff(d3f(1),z);
d4f(2) = diff(d3f(2),x) + diff(d3f(2),y) + diff(d3f(2),z);
d4f(3) = diff(d3f(3),x) + diff(d3f(3),y) + diff(d3f(3),z);
d4f = transpose(temp) * d3f;


syms x y z unreal
```

```
function [first, second, third, fourth] = EVAL(f, d2f, d3f, d4f, x0)

syms x y z real


first(1) = subs(f(1),x,y,z,x0(1),x0(2),x0(3));
first(2) = subs(f(2),x,y,z,x0(1),x0(2),x0(3));
first(3) = subs(f(3),y,x0(2));


second(1) = subs(d2f(1),x,y,z,x0(1),x0(2),x0(3));
second(2) = subs(d2f(2),x,y,z,x0(1),x0(2),x0(3));
second(3) = subs(d2f(3),x,y,z,x0(1),x0(2),x0(3));


third(1) = subs(d3f(1),x,y,z,x0(1),x0(2),x0(3));
third(2) = subs(d3f(2),x,y,z,x0(1),x0(2),x0(3));
third(3) = subs(d3f(3),x,y,z,x0(1),x0(2),x0(3));


fourth(1) = subs(d4f(1),x,y,z,x0(1),x0(2),x0(3));
fourth(2) = subs(d4f(2),x,y,z,x0(1),x0(2),x0(3));
fourth(3) = subs(d4f(3),x,y,z,x0(1),x0(2),x0(3));


syms x y z unreal
```

References

[1] Allen, N. A. (2005). *Computational Software for Building Biochemical Reaction Network Models with Differential Equations* (Doctoral dissertation, Virginia Polytechnic Institute and State University, 2001). Virginia Tech Digital Archives.

[2] Amgen, Inc. (2006). *Reversible Chemical Reaction Networks*. Unpublished manuscript. Thousand Oaks, California: Gilles Gnacadja.

[3] Baeza Baeza, J. J., Pérez Plá, F., and Ramis Ramos, G. (1992). "On the Integration of Kinetic Models Using a High-Order Taylor Series Method." *Journal of Chemometrics*. Vol 6, 231-246.

[4] Baeza Baeza, J. J., Pérez Plá, F., and Ramis Ramos, G. (1992). "Stiffness-Adaptive Taylor Method for the Integration of Non-Stiff and Stiff Kinetic Models." *Journal of Computational Chemistry*. Vol 13, Issue 7, 810-820.

[5] Bradie, B. (2004). DS9 trials and tribble-ations review. Retrieved October 8, 1997, from Psi Phi: Bradley's Science Fiction Club Web site: http://www.bradley.edu/campusorg/psiphi/DS9/ep/ 503r.html

[6] Butcher, J. C. (2003). *Numerical Methods for Ordinary Differential Equations*. Chichester, West Sussex, England: Wiley.

[7] Cornish-Browden, A. and Hofmeyr, JH. (2002). "The Role of Stoichiometric Analysis in Studies of Metabolism: An Example." *Journal of Theoretical Biology*. Vol 216

[8] Crampin, E. J., Schnell, S., and McSharry, P. E. (2004). "Mathematical and Computational Techniques to Deduce Complex Biochemical Reaction Mechanisms." *Progress in Biophysics & Molecular Biology*. Volume 86, 77-112.

[9] Famelis, I. Th. and Papakostas, S. N. (2004). "Symbolic Derivation of Runge-Kutta Order Conditions." *Journal of Symbolic Computation*. Volume 37 Issue 3, 311-327.

[10] Feinberg, M. (1995). "The Existence and Uniqueness of Steady States for a Class of Chemical Reaction Networks." Archive for Rational Mechanics and Analysis: Springer-Verlag. Volume 132, 311-370.

[11] Gavalas, G. (1968). *Nonlinear Differential Equations of Chemically Reacting Systems*. Berlin: Springer-Verlag.

[12] Golub, G. H. and Van Loan, C. (1996). *Matrix Computations*. (3rd Ed.) Baltimore: John Hopkins University Press.

[13] Gunawardena, J. (2003). "Chemical Reaction Network Theory for *in-silico* Biologists." Harvard University: Bauer Center for Genomics Research.

[14] Hairer E., Nørsett S.P. and Wanner G. (1993). *Solving Ordinary Differential Equations I.* (2nd Ed.) Berlin: Springer-Verlag.

[15] Hairer E. and Wanner G. (1996). *Solving Ordinary Differential Equations II.* (2nd Ed.) Berlin: Springer-Verlag.

[16] Hosea, M. E. and Shampine, L. F. (1996). "Analysis and Implementation of TR-BDF2." *Applied Numerical Mathematics*. Volume 20, 21-37.

[17] Koudriavtsev, A. B. and Jameson, R. F. (2001). *The Law of Mass Action*. Moscow: Springer-Verlag.

[18] Kincade, R. and Cheney, W. (2002). *Numerical Analysis: Mathematics of Scientific Computing*. Pacific Grove, California: Brooks/Cole.

[19] Lambert, J. D. (1991). *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. Chichester, West Sussex, England: Wiley.

[20] Mazkewitsch, D. (1963). "The n-th Derivative of a Product." *The American Mathematical Monthly*. Volume 70, Number 7, 739-742.

[21] Moler, C. and Van Loan, C. F. (2003). "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later." *Society for Industrial and Applied Mathematics (SIAM) Review*. Volume 45 Number 1, pp. 3000.

[22] Reverter, F., and Oller J. M. (1997). "A Modified Taylor Series Method for Solving Initial-Value Problems in Ordinary Differential Equations." *International Journal of Computer Mathematics*. Volume 65 Issue 3-4, 231-246.

[23] Shampine, L. F. and Reichelt, M. W. (1997). "The MATLAB ODE Suite." *SIAM Journal on Scientific Computing*. Volume 18, 1-22.

[24] H. Schmidt, M. Jirstrand. (2006). "Systems Biology Toolbox for MATLAB: A Computational Platform for Research in Systems Biology." *Bioinformatics.* Volume 22 Number 4, 514515.

[25] Spijker, M. N. (1996). "Stiffness in Numerical Initial-Value Problems." *Journal of Computational and Applied Mathematics*. Volume 72, 393-406.

[26] Steinfeld, J., Francisco, J, and Hase, W. (1989). *Chemical Kinetics and Dynamics*. New Jersey: Prentice Hall.

[27] Thaheem, A. B. and Laradji, A. (2003). "A Generalization of Leibniz Rule for Higher Derivatives." *International Journal of Mathematical Education in Science and Technology.* Volume 34, Number 6, 905-907.

[28] Tóth, J. and Li, G. (1997). "The Effect Of Lumping And Expanding On Kinetic Differential Equations." *SIAM Journal on Applied Mathematics*. Volume 57, Number 6, 1531-1556.

[29] Van Loan, C. F. (1975). "A Study of the Matrix Exponential." *Numerical Analysis Report Number 7*, University of Manchester, Manchester, UK.

[30] Warner, D. (1977). "The Numerical Solution of the Equations of Chemical Kinetics." *The Journal of Physical Chemistry.* Volume 81, Number 25, 2329.