Mathematical Models for the Kinetics of some Carbohydrate Enzymes

by

Vinh Quang Mai

A PhD thesis submitted to the
School of Mathematics, Statistics and Applied Mathematics,
National University of Ireland, Galway

Head of School: Dr. Rachel Quinlan

Head of Discipline: Dr. Martin Meere

Supervisors: Dr. Tuoi Vo and Dr. Martin Meere

Galway, Ireland
October 2019

Contents

List of Figures

List of Tables

Declaration

I declare that the work presented in this thesis is my own, and has not been previously submitted for award at another degree granting institution.

Signed: ________________________ ID: 15233989

Vinh Quang Mai

Date: ________________________

Abstract

In this thesis, some mathematical models for the mechanism of action of carbohydrate enzymes are developed and analysed.

Chapter 1 provides a general introduction to enzymes and their applications. In Chapter 2, mathematical preliminaries required for the subsequent analysis are introduced and discussed. This chapter also describes the Python software employed in the thesis.

In Chapter 3, a mathematical model for the degradation of hyaluronan by *Streptococcus pneumoniae* hyaluronate lyase is developed and analysed. The model results were found to agree well with experimental data. A Sobol global sensitivity analysis was implemented to identify the key model parameters. Some practical applications of the model are also indicated.

Chapter 4 considers a mathematical model that describes the phosphorylation of glucose by human hexokinase I. Numerical simulations of the model produce results that are consistent with the experimentally observed behaviour. A global sensitivity analysis of the model was implemented to help identify the key mechanisms of hexokinase I regulation. The sensitivity analysis also enabled the development of a simpler model that produces output close to that of the full model.

The model developed in Chapter 4 is too complex to obtain simple analytic expressions for the rate of product formation. It is also difficult to obtain simple qualitative insights into the enzyme behaviour from this model. In an effort to overcome those deficiencies, two simpler related models are developed in Chapter 5. The first model focuses on the mechanism of competitive product inhibition only, while the second model considers allosteric inhibition only. Some practical applications of the models are also indicated.

In Chapter 6, a brief discussion of the work presented in the thesis is given, and some possible directions for refinement and expansion of the models are also indicated.

Acknowledgements

Firstly, I thank my parents, brother, sisters, extended family, and my wife for their unconditional and invaluable encouragement and support over the past four years.

My supervisors, Dr. Martin Meere and Dr. Tuoi Vo, have been outstanding advisors, and I thank them for their timely and invaluable advice and help.

The School of Mathematics, Statistics and Applied Mathematics has provided an excellent environment in which to study and work as a researcher, and I am thankful to the academic and administrative staff for fostering such a pleasant atmosphere. My fellow postgraduate students have been a constant source of friendship for which I am thankful. I have been fortunate to share an office with many nice friends and have benefited from such a friendly work environment.

The members of my graduate research committee, Prof. Michel Destrade, Dr. Giuseppe Zurlo and Dr. Yury Rochev, have provided many helpful discussions and constant encouragement throughout the PhD programme.

My close friend, Dr. Thai Anh Nhan, has given me a lot of help and encouragement for which I am truly appreciative.

I am so grateful to colleagues in Thu Dau Mot University, especially Associate Professor, Dr. Hiep Van Nguyen and Dr. Ngan Kim Thi Nguyen, for their support.

Finally, I thank the College of Science at NUI Galway and Thu Dau Mot University for their financial support.

Chapter 1

Introduction

This thesis is concerned with developing mathematical models that describe the kinetics of carbohydrate enzymes. The principle enzymes of interest here are *Streptococcus pneumoniae* hyaluronate lyase and human Hexokinase I. In the current chapter, some general discussion of enzymes, their classification and industrial applications is provided. We also give some discussion of mechanisms of enzyme action, enzyme inhibition, as well as providing a general introduction to carbohydrate enzymes. We conclude the chapter with a thesis outline.

1.1 Introduction to enzymes

Enzymes are biological catalysts that are naturally occurring in most living organisms. In 1833, the French chemist Anselme Payen found the first enzyme - diastase [12]. In 1877, the German physiologist Wilhelm Kühne first used the word "enzyme" to describe the ability of yeast to produce alcohol from sugars [13, 3].

Most enzymes are proteins, except for a few enzymes that are composed of ribonucleic acids or ribonucleoproteins. The active site of an enzyme is where substances bind and where a reaction is catalysed to produce a new compound. An individual enzyme is typically able to catalyse a few specific reactions [14, 15, 16, 17, 18, 19]. Enzymes are much larger than their substrates, and their molecular weights range from 10,000 - 2,000,000 Dalton [15]. Figure 1.1 shows a Hexokinase I enzyme molecule of *Kluyveromyces lactis* in crystal form, and its ligands.

Enzymes are effective catalysts in the sense that they are capable of greatly accelerating biochemical reaction rates even when the enzyme is at very low concentrations. Furthermore, enzymes are not consumed during the reactions, and this is one of the commercial advantages of enzymes. In order for a reaction to take place, an amount of energy, called the activation energy, is needed irrespective of whether the reaction consumes or releases energy. Enzymes speed up reactions by reducing the required activation energy; see Figure 1.2.

Enzyme-catalysed reactions are frequently represented as follows

$$\text{Substrate} \xrightleftharpoons{\text{Enzyme}} \text{Product}.$$

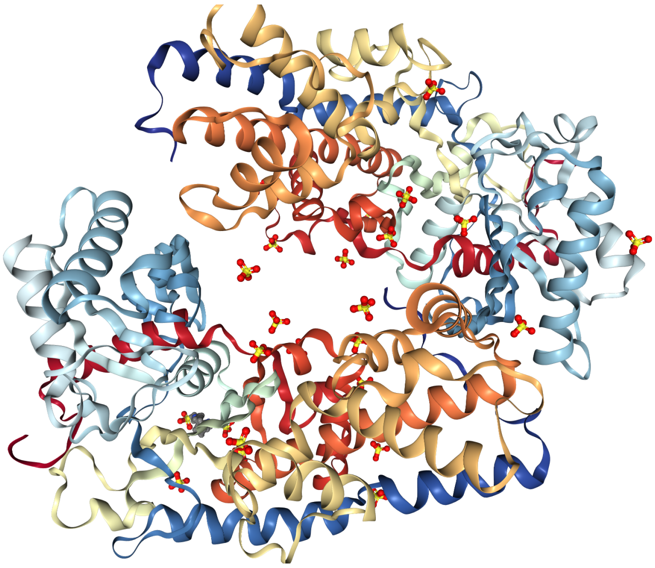


Figure 1.1: Crystal structure of a dimeric Hexokinase I enzyme molecule from *Kluyveromyces lactis*. In the diagram, ligands of the enzyme are depicted by balls and sticks. Extracted from [1, 2], PDB: 3O08.

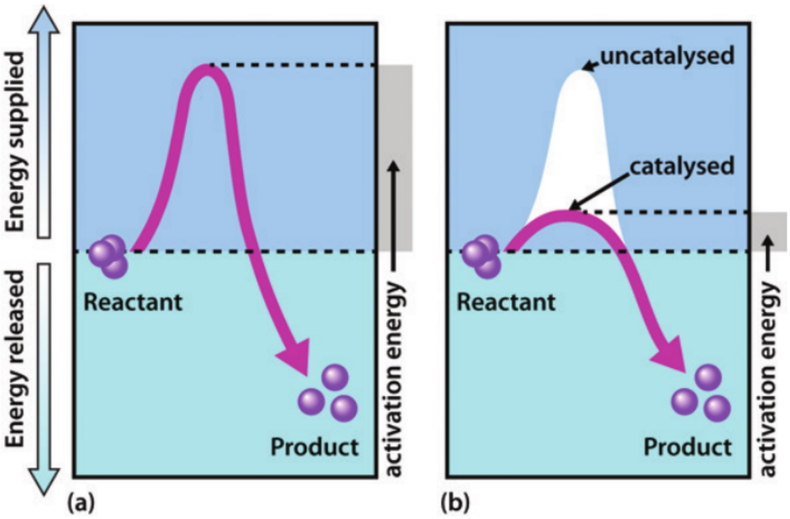


Figure 1.2: An enzyme reduces the activation energy required to initiate a chemical reaction. (a) represents an uncatalysed reaction, and (b) represents an enzyme-catalysed reaction. Extracted from [3].

Enzymes play an important role in metabolism because of their ability to catalyse biochemical reactions at typical biological temperatures and pH levels.

Enzyme-catalysed reactions take place at much faster rates than reactions without enzyme [19, 20]. Chemical reactions underlie the chemical basis of life, and enzymes are a key factor in the metabolism of cells [14, 15]. Currently, approximately 1300 different enzymes have been found in human cells [21], so it is useful to give some discussion of the nomenclature and classification of enzymes. This is provided in the next section.

1.2 Nomenclature and classification

By international convention, there are seven classes of enzymes and these are distinguished by the type of chemical reactions that they catalyse. Each class can be divided into subclasses based on the nature of the chemical groups and coenzymes involved in their reactions. A coenzyme is an organic or metalloorganic molecule that works with an enzyme to initiate or aid the function of that enzyme [16, 19]. The seven classes of enzymes are [8, 9]:

- *Class 1: Oxidoreductases.* Enzymes in this class transfer hydrogen atoms or oxygen atoms or electrons from one substrate to another. These enzymes includes the dehydrogenases, reductases, oxidases, dioxygenases, hydroxylases, peroxidases, and catalases.

- *Class 2: Transferases.* These enzymes transfer chemical groups between substrates, and they include the kinases, aminotransferases, acetyltransferases, and carbamoyltransferases.

- *Class 3: Hydrolases.* Enzymes in this class catalyse the hydrolytic cleavage of bonds, and they include the peptidases, esterases, phosphatases, and sulphatases.

- *Class 4: Lyases.* These enzymes catalyse elimination reactions that result in the formation of double bonds. Adenylyl cyclase, enolase and aldolase are lyases.

- *Class 5: Isomerases.* Enzymes in this class interconvert isomers of various types by intramolecular rearrangements. They include phosphoglucomutase and glucose-6-phosphate isomerase.

- *Class 6: Ligases (also called synthases).* These enzymes catalyse covalent bond formation with the concomitant breakdown of a nucleoside triphosphate, commonly ATP. Carbamoyl phosphate synthase and DNA ligase are examples of ligases.

- *Class 7: Translocases.* These enzymes catalyse the movement of ions or molecules across membranes or their separation within membranes. They include Na(+)-transporting two-sector ATPase and ABC-type polar-amino-acid transporter [22, 23].

According to the Enzyme Commission (EC) rules, each enzyme is given a unique code of four digits and an obvious systematic name based on the reaction it catalyses. The nomenclature of an individual enzyme consists of the letters "EC" followed by four digits separated by points [24].

- The first digit determines the general type of reaction the enzyme catalyses and ranges from one to seven, corresponding to the seven categories described above.
- The second digit describes the subclass.
- The third digit indicates the sub-subclass.
- The fourth digit is the serial number of the enzyme in its sub-subclass.

It should be noted that subclasses of different classes are different even though they are assigned the same number; see [8] for more details. Fortunately, it is now easy to find this information for any individual enzyme using the Enzyme Nomenclature Database (available at `https://enzyme.expasy.org/`). For example, the Enzyme Commission Number for the hyaluronate lyase enzyme is EC 4.2.2.1. Here the 4 represents the group Lyases. The 2 indicates the subclass Carbon-oxygen lyases, and the second 2 gives the sub-subclass Acting on polysaccharides - this means that this sub-subclass catalyses polysaccharides. Finally, the 1 is the serial number for the hyaluronate lyase enzyme [25]. A summary of the seven principle classes of enzymes is given in Table 1.1.

Table 1.1: Enzyme classification. The main classes of enzymes in the EC classification system [3, 8, 9].

First EC digit	**Enzyme class**	**Reaction type**
1.	Oxidoreductases	Oxidation/reduction
2.	Transferases	Atom/group transfer (excluding other classes)
3.	Hydrolases	Hydrolysis
4.	Lyases	Group removal (excluding 3.)
5.	Isomerases	Isomerisation
6.	Ligases	Joining of molecules linked to the breakage of a pyrophosphate bond
7.	Translocases	Movement of ions or molecules across membranes or their separation within membranes

Enzymes accelerate reactions even at very low concentrations and are not consumed by the reactions. These features of enzymes provide commercial, sustainable and environmental advantages. In the next section, we give some discussion of industrial applications of enzymes.

1.3 Industrial applications of enzymes

As mentioned above, enzymes offer a variety of benefits. Nowadays, scientific and technological advances facilitate the study of enzymes and their applications [26, 27]. Increasingly, new enzymes are being extracted and studied. Numerous applications of enzymes have been investigated and developed in biotechnology, industry, and medicine [3, 10, 28, 29, 30]. In this section, some common applications of enzymes are described; see Table 1.2.

Table 1.2: Industrial applications of enzyme catalysis [10].

Sector	**Enzymes**	**Applications**
Pharmaceuticals	Nitrile hydratase, transaminase, monoamine oxidase, lipase, penicillin acylase	Synthesis of intermediates for production of active pharmaceutical ingredients
Food Processing	Trypsin, amylase, glucose isomerase, papain, pectinase	Conversion of starch to glucose, production of high fructose corn syrup, production of prebiotics, debittering of fruit juice
Detergent	Protease, lipase, amylase, cellulase	Stain removal, removal of fats and oils, color retention
Biofuels	Lipase, xylanase, cellulase	Production of fatty acid methyl esters, decomposition of lignocellulotic material for bioethanol production
Paper and Pulp	Lipase, cellulase, xylanase	Removal of lignin for improved bleaching, improvement of fiber properties

Before an application for an enzyme can be developed, its mechanism of action must first be understood. Besides experimental studies, mathematical models are also helpful tools to help gain insights into the mechanism of action of an enzyme. Hence, some discussion of the theoretical tools employed to analyse enzymes is appropriate, and this is provided in the next section.

1.4 Mechanism of enzyme action

Enzymes speed up chemical reactions by reducing the activation energy required to initiate reactions. In enzymatic reactions, there exists at least one substance, called the substrate, that is converted to another substance, called the product. How do enzyme and substrate molecules bind to make a reaction take place? Recall that enzymes are macromolecules that are typically much larger than their substrates. Two theoretical models have been proposed for the binding of a substrate molecule to an enzyme molecule; the "lock and key" model, and the induced-fit model. In the induced-fit model, the enzyme makes conformational changes during binding to form a precise fit with its substrate using multiple weak interactions and hydrophobic characteristics on the enzyme surface mold [31]. The right hand side of Figure 1.3 depicts an example of an induced-fit.

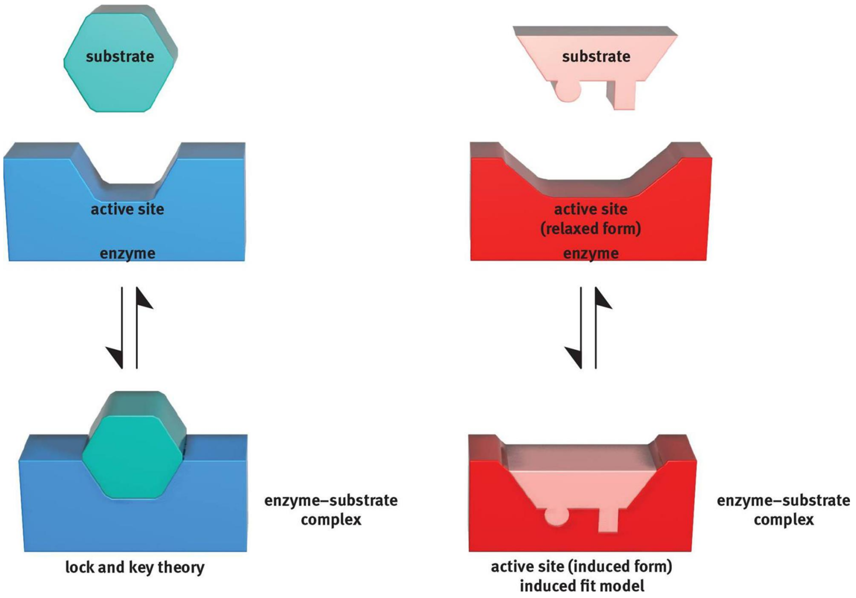


Figure 1.3: The lock and key mechanism for enzymes (left). The induced fit mechanism for enzymes (right). Extracted from [4].

In the "lock and key" model, the shape of the active site of an enzyme molecule is complementary to that of its substrate. The exact fit of the substrate to the active site of the enzyme is akin to the fit of a key into a lock [32]. The "lock" here refers to enzyme, and the "key" refers to its substrate. The left hand side of Figure 1.3 depicts the lock and key mechanism.

Using the lock and key model, the kinetic mechanism of an enzymatic reaction can be written as follows

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_0} E + P, \tag{1.1}$$

where E denotes the enzyme, S the substrate, ES the enzyme-substrate complex, and P the product. The constant k_1 is the adsorption constant rate, k_{-1} the desorption constant rate, and k_0 the catalytic or turnover constant rate. Figure 1.4 depicts a simple enzymatic reaction. Using the reactions given in (1.1), the well-known Michaelis-Menten formula for the rate of product formation of enzyme-catalysed reactions may be derived using elementary arguments. This is given by

$$v = \frac{V_{max}[S]}{[S] + K_m}, \tag{1.2}$$

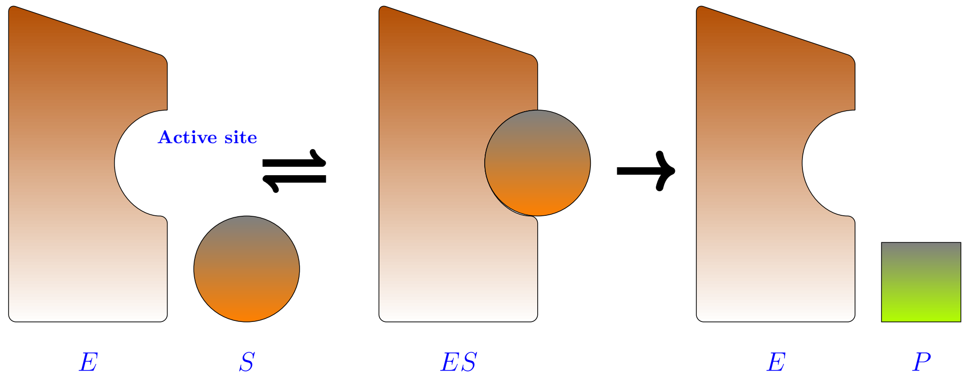


Figure 1.4: A schematic depiction of a simple enzymatic reaction. E denotes the enzyme, S the substrate, ES the enzyme-substrate complex, and P the product.

where

$$V_{max} = k_0 e_0, \quad K_m = \frac{k_{-1} + k_0}{k_1},$$

$[S]$ is the substrate concentration, and e_0 is the initial free enzyme concentration. Here K_m and V_{max} are referred to as the Michaelis constant of the enzyme and the maximal rate of the reaction, respectively. The maximal rate V_m gives the maximal rate of product formation. The Michaelis constant K_m gives the concentration of substrate for which the product formation rate is half the maximal rate. The formula (1.2) is extensively used in the investigation of enzymes.

In reality, the kinetic mechanism of an enzyme is usually complicated. For example, the Bi Bi random mechanism is a kinetic mechanism in which the enzyme catalyses two substrates to form two products. This mechanism arises in Chapter 4 of this thesis as a model for hexokinase I enzyme. It may be represented as follows

$$\begin{tikzcd}
 & EA \arrow[dr, dash] & & & EC \arrow[dr, dash] & \\
E \arrow[ur, dash] \arrow[dr, dash] & & EAB \arrow[r, shift left, harpoon] & ECD \arrow[l, shift left, harpoon] \arrow[ur, dash] \arrow[dr, dash] & & E, \\
 & EB \arrow[ur, dash] & & & ED \arrow[ur, dash] &
\end{tikzcd} \tag{1.3}$$

where E is the enzyme; A, B are the substrates; C, D are the products; and EAB, ECD are the enzyme-substrates and enzyme-products complexes, respectively. In equation (1.3), an A substrate molecule and a B substrate molecule randomly bind to a free enzyme molecule E to form an enzyme-substrate complex EAB which is then reversibly catalysed to form an enzyme-product complex ECD. The enzyme molecule randomly releases the product molecules C and D to restore the free enzyme.

Enzymes reside in living cells and speed up cellular metabolic chemical reactions, enabling the rapid production of cellular metabolites. In cells, biochemical substances are required to show up at the right time and at the right concentration. Hence, there are many mechanisms to regulate the activities of enzymes in cells, and one particularly important regulatory mechanism is enzyme inhibition. Some discussion of enzyme inhibition is given in the next section.

1.5 Enzyme inhibition

Enzymes are responsible for many biochemical reactions involved in the metabolism of cells. Cells strictly regulate the activities of enzymes through activation and inhibitory mechanisms. We shall consider inhibitory mechanisms here because these will feature in future chapters. Substances that can bind to the enzyme to interfere with the catalytic action of the enzyme are called enzyme inhibitors. Consequently, the presence of inhibitors slows down catalysis and can in some cases stop catalysis altogether. There are three common types of enzyme inhibition: competitive, non-competitive, and substrate inhibition. We restrict our attention here to competitive and non-competitive inhibition.

In competitive inhibition, the inhibitor competes with the substrate for the active site of the enzyme. In non-competitive inhibition, the inhibitor does not compete with the substrate for the active site, but rather binds to a distinct site of the enzyme. This binding alters the shape of its active site so that the substrate can no longer bind to the enzyme. In any inhibition, the rate of the enzymatic reaction reduces with the increasing concentration of inhibitor [18, 33, 34]. Figure 1.5 depicts competitive inhibition (left), and non-competitive inhibition (right).

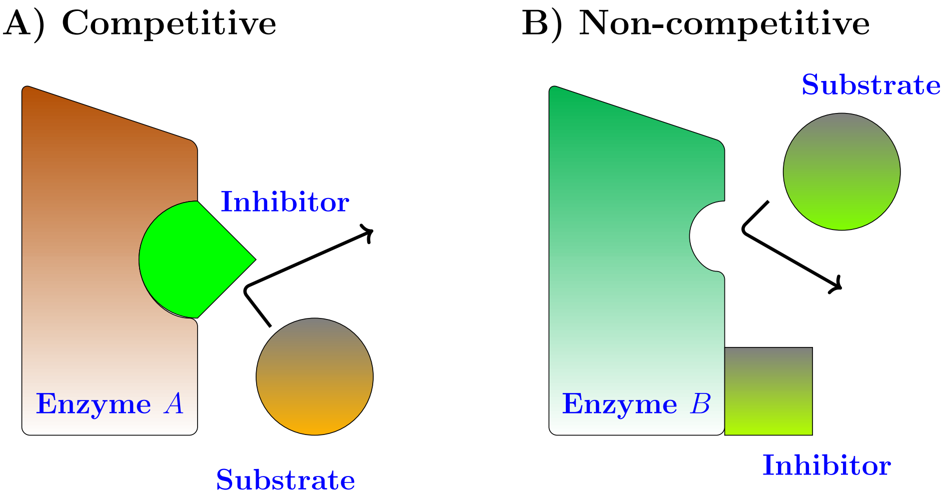


Figure 1.5: A) Competitive inhibition, and B) Non-competitive inhibition.

Carbohydrate enzymes have carbohydrates as their substrates. Carbohydrate enzymes belong to a number of different classes, for example, hyaluronate lyase belongs to the Class 4 Lyases [25], while Hexokinase belongs to the Class

2 Transferases [35]. Carbohydrate enzymes form the topic of the current study, and so we provide an overview of carbohydrate enzymes in the next section.

1.6 Carbohydrate enzymes

Carbohydrates, proteins, lipids, and nucleic acids are the four major types of organic molecules of living systems [36]. Carbohydrates are the most plentiful organic molecules found in nature, and the synthesis and metabolism of carbohydrates takes place in nearly all organisms [37]. The fact that the empirical formula of most simple sugars is $C_nH_{2n}O_n$ $(n \geq 3)$, implies that carbon atoms are associated in some way with water. These compounds are referred to as "hydrates of carbon" or "carbohydrates" [19, 37]. Most nonphotosynthetic cells produce energy by oxidising carbohydrates [19].

Carbohydrates can occur as monosaccharides, oligosaccharides, and polysaccharides. Monosaccharides include a single polyhydroxy aldehyde or ketone unit. The six-carbon sugar D-glucose is the most abundant monosaccharide found in nature. Oligosacchrides are short chains of monosaccharide units linked together by glycosidic bonds. The most abundant oligosaccharides in nature are disaccharides. Polysaccharides are sugar polymer chains of more than 20 monosaccharide units. For example, cellulose is a linear polysaccharide of one monosaccharide type [38], while hyaluronan is a linear polysaccharide with two monosaccharide types [39, 40]. Glycogen is a multibranched polysaccharide with glucose units [19, 41, 42, 43].

Carbohydrates play important roles in the metabolism of most living organisms. For examples, glucose is a major source of energy for living organisms, and hyaluronan, a high molecular weight polysaccharide, is a constituent of the extracellular matrix of cells [40]. Carbohydrate enzymes appear in many forms, and carry out a range of functions in the body, including biosynthesis, modification, binding, and catabolism of carbohydrates. They are classified into six families, as follows: [44].

- Glycosyltransferases [45, 46, 47],
- Glycoside Hydrolases [48, 49, 50],
- Polysaccharide Lyases [51, 52],
- Carbohydrate Esterases [53, 54],
- Auxiliary Activity Families [55],
- Carbohydrate Binding Modules (non-catalytic; included due to their association with catalytic modules) [56].

Figure 1.6 shows a ribbon diagram for a molecule of human hyaluronidase I; this enzyme is a member of the Glycoside Hydrolases.

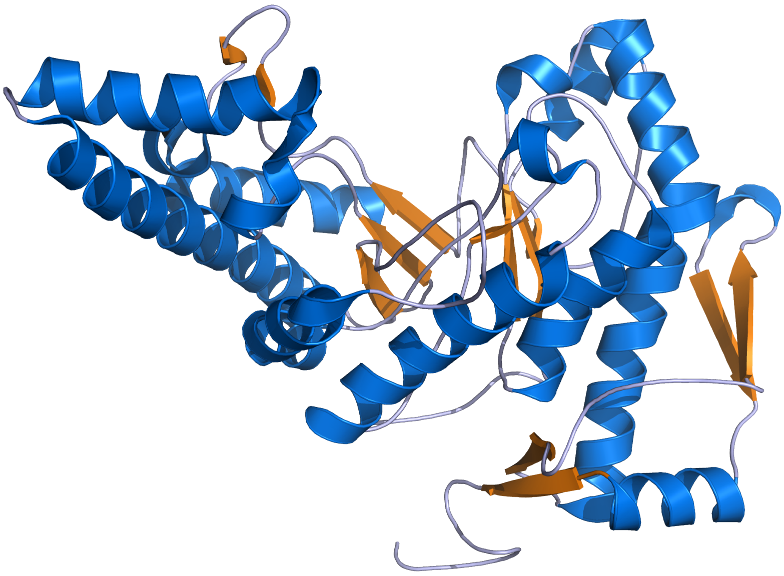


Figure 1.6: Ribbon diagram for a human hyaluronidase I molecule. This structure was created using PyMOL [5]. Extracted from [6].

The importance of enzymes in living organisms, as well as their numerous applications, motivated the current study. We shall develop mathematical models describing specific enzymatic reactions with a view to obtaining practically useful insights into their behaviour. The next section presents an overview of the thesis.

1.7 Thesis outline

This project is concerned with the mathematical modelling of the kinetics of carbohydrate enzymes. Mathematical modelling is a useful tool to help understand the kinetic mechanisms of enzymes. Moreover, model results may assist with the design of experiments and suggest further applications.

In Chapter 1, the subject area is introduced by giving some discussion of enzymes and their applications.

In Chapter 2, some mathematical preliminaries that are required for the subsequent chapters are discussed. In particular, we discuss the Law of Mass Action, parameter estimation, and global sensitivity analyses. A brief discussion of scientific Python packages is also provided.

In Chapter 3, a mathematical model that describes the degradation of hyaluronan by *Streptococcus pneumoniae* hyaluronate lyase is considered. In this system, (i) enzyme randomly binds to a hyaluronan polymer chain. (ii) The enzyme then degrades the hyaluronan chain to form two shorter chains, and releases the chain with an unsaturated end. (iii) The enzyme translocates along the truncated chain by one disaccharide unit torward the non-reducing end to recover the original bound state. The process then repeats (move back up to step (ii)) until the remaining chain is fully degraded, with all subsequent degradation products being single unsaturated disaccharide units. Our theoretical results are not only consistent with experimental data for the enzyme, but also well agree with experimental data for other bacterial hyaluronidases.

In Chapter 4, a mathematical model that describes the phosphorylation of glucose by human hexokinase I is investigated. In this system, hexokinase I enzyme transfers a phosphate group of an ATP molecule to a glucose molecule to produce an ADP molecule and a glucose-6-phosphate. Glucose-6-phosphate inhibits the enzyme using both competitive and allosteric inhibitory mechanisms. Inorganic phosphate antagonises the inhibition of the enzyme by glucose-6-phosphate for low concentrations, and inhibits the enzyme for high concentrations. Our theoretical results are consistent with the experimentally observed behaviours.

In Chapter 5, two mathematical models (model I and model II) that describe the kinetics of enzymes with product inhibition are studied. Model I describes enzymes subject to competitive product inhibition, while Model II describes enzymes subject to allosteric product inhibition. A formula for the rate of product formation is found for each model. Model I can be used to obtain, under certain circumstances, the rate of phosphorylation of glucose by mini hexokinase I. Model II can be used to obtain, in certain circumstances, the rate of phosphorylation of glucose by a mutant hexokinase I.

Finally, in Chapter 6, some brief discussion of the work presented in the thesis is presented. It summarises the model results and suggests possible directions for refinement and expansion of the models.

Chapter 2

Mathematical preliminaries

In this chapter, we describe the principal mathematical tools used in the remainder of the thesis. The theme of the thesis is the construction and analysis of mathematical models describing the action of carbohydrate enzymes. The mathematical models will consist of coupled systems of nonlinear ordinary differential equations, and will be formulated using the principle of mass action. Hence, we give some discussion below of the concepts underlying mass action.

The models we shall develop have quite a large number of parameters, and values will need to be assigned to each of these to perform numerical simulations. Where possible, parameter values are obtained from the literature. Another approach is to estimate parameter values using parameter estimation techniques in conjunction with experimental data. Hence, we shall also discuss parameter estimation techniques.

Nevertheless, there will inevitably be some uncertainty in the values of the parameters estimated, and so it is of value to use a sensitivity analysis to assess the sensitivity of the model output to variations in the model parameters. Hence, we also describe sensitivity indices and a global sensitivity analysis. A section describing the Python software used to numerically integrate the differential equations is also provided.

2.1 The law of mass action

In this section, we give some discussion of the law of mass action [33, 57, 58, 59]. Consider the following chemical reaction

$$A + B \to C, \tag{2.1}$$

where A and B are the reactants, and C is the reaction product. The reaction (2.1) states that one molecule of A and one molecule of B are required to form one molecule of C. It follows that the rate of change of the concentrations of

A and B are the same, and they are equal to the negative of the rate of change of C. Hence, we have

$$\frac{d[A]}{dt} = \frac{d[B]}{dt} = -\frac{d[C]}{dt},$$

where $[X]$ denotes the concentration of the reactant X. In other words,

$$\begin{aligned} \frac{d[A]}{dt} &= -v, \\ \frac{d[B]}{dt} &= -v, \\ \frac{d[C]}{dt} &= v, \end{aligned} \tag{2.2}$$

where v is called the rate of the reaction. Now v depends on the collision frequency of A and B, and this implies that it depends on $[A]$ and $[B]$. Clearly, collisions do not take place if A or B is absent, so the rate is then zero. Hence, we are assuming that $v = v([A], [B])$, where $v([A], 0) = v(0, [B]) = 0$. Using Taylor's theorem, we obtain an approximation of this function as follow

$$\begin{aligned} v = v(0,0) + \frac{\partial v}{\partial [A]}(0,0)[A] + \frac{\partial v}{\partial [B]}(0,0)[B] + \frac{1}{2}\frac{\partial^2 v}{\partial [A]^2}(0,0)[A]^2 \\ + \frac{\partial^2 v}{\partial [A] \partial [B]}(0,0)[A][B] + \frac{1}{2}\frac{\partial^2 v}{\partial [B]^2}(0,0)[B]^2 + \cdots . \end{aligned} \tag{2.3}$$

Recall that $v([A], 0) = v(0, [B]) = 0$, so that

$$\frac{\partial v}{\partial [A]}([A],0) = \frac{\partial^2 v}{\partial [A]^2}([A],0) = \frac{\partial v}{\partial [B]}(0,[B]) = \frac{\partial^2 v}{\partial [B]^2}(0,[B]) = 0.$$

It follows that the first nonzero term in (2.3) is

$$\frac{\partial^2 v}{\partial [A] \partial [B]}(0,0)[A][B] \neq 0,$$

and we thus have

$$v \approx k[A][B], \tag{2.4}$$

where k is known as the rate constant and we have neglected terms higher than order two. This expression, along with the rate equations (2.2), is the Law of Mass Action as applied to the reaction (2.1). We now list the assumptions underlying the Law of Mass Action.

Definition 2.1.1 ([57]). The Law of Mass Action consists of the following three assumptions:

1. The rate, v, of the reaction is proportional to the product of the reactant concentrations, with each concentration raised to the power of its stoichiometric coefficient.

2. The rate of change of the concentration of each species in the reactions is the product of its stoichiometric coefficient with the rate of the reaction, adjusted for sign (+ if product and - if reactant).

3. For a system of reactions, the rates add.

To illustrate the ideas above, we consider the following example.

Example 2.1.1. Consider a system of reactions defined by

$$\begin{aligned} A + 2B &\overset{k_1}{\to} C, \\ C &\overset{k_2}{\to} A + 2B, \end{aligned}$$

where k_1 and k_2 are the rate constants for the reactions. Using the rules described above, the governing equations of the model describing the system are given by

$$\begin{aligned} \frac{d[A]}{dt} &= -k_1[A][B]^2 + k_2[C], \\ \frac{d[B]}{dt} &= -k_1[A][B]^2 + 2k_2[C], \\ \frac{d[C]}{dt} &= -k_2[C] + k_1[A][B]^2. \end{aligned}$$

Conservation laws are sometimes useful for reducing the number of kinetic equations appearing in a system. Conservation laws are defined as follows.

Definition 2.1.2 ([57])**.** Given species concentrations $[A]$, $[B]$, $[C]$, ..., $[Z]$ (functions of time t) and numbers a, b, c, ..., z then

$$a[A] + b[B] + c[C] + \cdots + z[Z]$$

is said to be conserved if

$$\frac{d}{dt}(a[A] + b[B] + c[C] + \cdots + z[Z]) = 0. \tag{2.5}$$

It is required that at least one of the numbers $a, b, c, ..., z$ is nonzero, and that (2.5) does not depend on the initial conditions and rate constants. The corresponding conservation law is then

$$a[A] + b[B] + c[C] + \cdots + z[Z] = constant.$$

Consider the model equations in Example 2.1.1. It is clear that

$$\frac{d[A]}{dt} + \frac{d[C]}{dt} = 0.$$

Hence, a conservation law for this system is

$$[A] + [C] = constant.$$

It should be noted that there may be many independent conservation laws for a system; see [57] for more details.

As mentioned above, the coefficient k in (2.4) is constant. In reality, k can depend on the conditions under which the reaction takes place [57]. For example, the rate of a chemical reaction can depend strongly on the temperature, and the rate of an enzymatic reaction can depend on both the temperature and the pH level. In the current study, it is assumed that the system of interest takes place in a medium that maintains the same conditions throughout. It is also assumed that the concentrations of species in the system are sufficiently high so that a probabilistic model is not required [59].

Using the Law of Mass Action and the kinetic mechanism of a system of interest, we can develop a model consisting of ordinary differential equations that describes the evolution of the concentrations of species in the system in time. Normally, parameters values for the model are not available in the literature. Another approach is to estimate the values of parameters with the aid of experimental data. In the next section, we discuss some parameter estimation techniques.

2.2 Parameter estimation

We now discuss methods to estimate parameters values in a model. A particular focus here is the method of least squares, a common approach in parameter estimation, and one that we will subsequently use in Chapter 3. In this method, the sum of squared residuals between model outputs and data is minimised using various optimisation algorithms, such as the Nelder-Mead method or the Sequential Least SQuares Programming method (SLSQP) [60, 61, 62, 63, 64].

To help fix ideas, we consider a specific example. In modelling biochemical systems, the mathematical structure of a model is usually known, while the parameter values remain to be determined. Let $y(p,t)$ be the model output that is a function of time t, and the single parameter p, and let y_1, y_2, ..., y_m be experimental data for the species of interest corresponding to the time points t_1, t_2, ..., t_m. To estimate value of the parameter p, the sum of squared residuals

$$\Theta = \sum_{i=1}^{m} (y(p,t_i) - y_i)^2 \tag{2.6}$$

will be minimised using algorithms available in the literature [60, 61, 62, 63, 64]. The SLSQP method uses Sequential Least SQuares Programming to minimise a function of one or several variables with any combination of bounds, equality and inequality constraints on the parameters [65]. Theoretical algorithms and optimisation software are available in the literature [58, 66, 67, 64, 65]. Below, we consider a simple example to illustrate this approach to parameter estimation.

Example 2.2.1. Given a model defined by

$$\frac{dy}{dt} = 2at + b, \quad y(0) = 0, \tag{2.7}$$

and experimental data given by

$$y_1 = y(1) = 0, \quad y_2 = y(2) = 2, \quad y_3 = y(4) = 13.$$

To estimate values of the model parameters, we solve the equation (2.7) to obtain a general solution as follows

$$y(t) = at^2 + bt + c, \tag{2.8}$$

where c is a constant. Substituting the initial condition into the equation (2.8) gives

$$y(0) = c = 0,$$

so that

$$y(t) = at^2 + bt. \tag{2.9}$$

Using (2.9) and (2.6), we now calculate the sum of squared residuals as follows

$$\Theta = \sum_{i=1}^{3}(y(t_i) - y_i)^2 = 273a^2 + 146ab - 432a + 21b^2 - 112b + 173 \tag{2.10}$$

where $t_1 = 1$, $t_2 = 2$, and $t_3 = 4$. To minimise this Θ, we calculate

$$\begin{aligned} \frac{\partial \Theta}{\partial a} &= 546a + 146b - 432 = 0, \\ \frac{\partial \Theta}{\partial b} &= 42b + 146a - 112 = 0, \end{aligned} \tag{2.11}$$

and solving these equations gives

$$a = \frac{112}{101}, \quad b = -\frac{120}{101}.$$

One can check that these values correspond to a minimum value of the Θ function. Figure 2.1 compares the solution $y(t)$ for the estimated parameter values with the experimental data.

It should be noted that analytical solutions to ODEs are not usually available in practice, and numerical techniques need to be resorted to. Therefore, it is rarely feasible to handle parameter estimation by direct calculation as in Example 2.2.1. Parameter estimation is in fact usually implemented using computational packages, such as the SciPy package for Python [68, 65]. It should also be emphasised that there is always uncertainty in the values of the parameters. Also, there are measurement errors in experimental data, such as instrument and operator errors. Sensitivity analyses assist with evaluating the

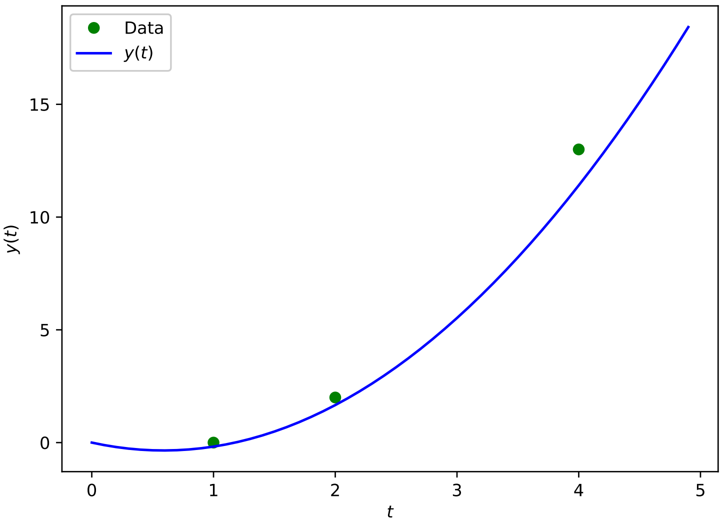


Figure 2.1: Comparison between a fitted theoretical curve and experimental data; see Example 2.2.1

significance of errors in parameters values. Also, numerous statistical methods have been developed to assess measurement errors; see [69, 70, 71, 72] for more details.

In summary, to implement parameter estimation for a model, we need to obtain analytically or numerically the solution to the model, and then create a sum of squared residuals of the solution and experimental data. The parameters are then estimated using an appropriate algorithm to minimise the sum of squared residuals.

Once values for the parameters have been obtained, it is instructive to evaluate how variations in these values affect the model output. This can be achieved by carrying out a global sensitivity analysis, and we discuss this next.

2.3 Global sensitivity analysis

To evaluate how variations in the values of the parameters affect model output, total, first-order, second-order, and higher-order sensitivity indices may be calculated [73]. Many theories and techniques for sensitivity analyses have been developed [73, 74, 75, 76, 77, 78, 79, 80]. In this section, we restrict my attention to a Sobol global sensitivity analysis. To simplify the discussion, we develop a sensitivity analysis for a model consisting of an ordinary differential equation.

Suppose that the model is governed by the ordinary differential equation

$$\frac{dy}{dt} = f(y, t, \mathbf{p}),$$

where $\mathbf{p} = (p_1, p_2, p_3)$ are the model parameters. Each parameter ranges over a finite interval which may be assumed, after rescaling, to be $[0,1]$. Let $I = [0,1]$, $I^2 = [0,1] \times [0,1]$, $I^3 = [0,1] \times [0,1] \times [0,1]$, and let $y(t, \mathbf{p})$ be the solution of the model. For $t = t_0$ fixed, we write $y(p) = y(t_0, \mathbf{p})$ for brevity. We now define new functions as follows [73]:

$$\begin{aligned}
y_0 &= \int\limits_{I^3} y(p) dp_1 dp_2 dp_3, \\
y_1(p_1) &= \int\limits_{I^2} y(p) dp_2 dp_3 - y_0, \\
y_2(p_2) &= \int\limits_{I^2} y(p) dp_1 dp_3 - y_0, \\
y_3(p_3) &= \int\limits_{I^2} y(p) dp_1 dp_2 - y_0, \\
y_{12}(p_1, p_2) &= \int\limits_{I} y(p) dp_3 - y_0 - y_1(p_1) - y_2(p_2), \\
y_{13}(p_1, p_3) &= \int\limits_{I} y(p) dp_2 - y_0 - y_1(p_1) - y_3(p_3), \\
y_{23}(p_2, p_3) &= \int\limits_{I} y(p) dp_1 - y_0 - y_2(p_2) - y_3(p_3), \\
y_{123}(p_1, p_2, p_3) &= y(p) - y_0 - y_1(p_1) - y_2(p_2) - y_3(p_3) \\
&\qquad - y_{12}(p_1, p_2) - y_{13}(p_1, p_3) - y_{23}(p2, p3).
\end{aligned} \tag{2.12}$$

Integrating each of the expressions on the left hand side of $(2.12)_2$ - $(2.12)_8$ over the domain I gives zero. For example,

$$\int\limits_{I} y_1(p_1) dp_1 = \int\limits_{I} \left(\int\limits_{I^2} y(p) dp_2 dp_3 - y_0 \right) dp_1 = \int\limits_{I^3} y(p) dp_1 dp_2 dp_3 - y_0 = 0,$$

or

$$\begin{aligned}
\int\limits_{I} y_{23}(p_2, p_3) dp_2 &= \int\limits_{I} \left(\int\limits_{I} y(p) dp_1 - y_0 - y_2(p_2) - y_3(p_3) \right) dp_2 \\
&= \int\limits_{I^2} y(p) dp_1 dp_2 - y_0 - y_3(p_3) = 0,
\end{aligned}$$

and so on. We have the following expansion for $y(p)$:

$$y(p) = y_0 + \sum_{i=1}^{3} y_i(p_i) + y_{12}(p_1, p_2) + y_{13}(p_1, p_3) + y_{23}(p_2, p_3) + y_{123}(p_1, p_2, p_3). \tag{2.13}$$

The right-hand side of (2.13) is called the ANOVA-representation of $y(p)$; see [73] for a discussion of this. ANOVA is an acronym for Analysis Of Variances [81].

Squaring (2.13) and integrating over I^3, gives

$$\begin{aligned}\int\limits_{I^3} y^2(p)dp_1dp_2dp_3 - y_0^2 = &\sum_{i=1}^{3}\int\limits_{I} y_i^2(p_i)dp_i + \int\limits_{I^2} y_{12}^2(p_1,p_2)dp_1dp_2\\ &+\int\limits_{I^2} y_{13}^2(p_1,p_3)dp_1dp_3 + \int\limits_{I^2} y_{23}^2(p_2,p_3)dp_2dp_3\\ &+\int\limits_{I^3} y_{123}^2(p_1,p_2,p_3)dp_1dp_2dp_3. \qquad (2.14)\end{aligned}$$

The constants

$$\begin{aligned}&D = \int\limits_{I^3} y^2(p)dp_1dp_2dp_3 - y_0^2, \quad D_i = \int\limits_{I} y_i^2(p_i)dp_i, \quad i=1,2,3,\\ &D_{12} = \int\limits_{I^2} y_{12}^2(p_1,p_2)dp_1dp_2 \qquad D_{13} = \int\limits_{I^2} y_{13}^2(p_1,p_3)dp_1dp_3, \qquad (2.15)\\ &D_{23} = \int\limits_{I^2} y_{23}^2(p_2,p_3)dp_2dp_3 \qquad D_{123} = \int\limits_{I^3} y_{123}^2(p_1,p_2,p_3)dp_1dp_2dp_3\end{aligned}$$

are called variances, and from (2.14) it follows that

$$D = D_1 + D_2 + D_3 + D_{12} + D_{13} + D_{23} + D_{123}. \qquad (2.16)$$

It is clear that if $\mathbf{p}$ were a random point uniformly distributed in I^3, with the parameters p_1, p_2, and p_3 mutually independent, then $y(p)$, y_1, y_2, y_3, y_{12}, y_{13}, y_{23}, and y_{123} would be random variables with corresponding variances D, D_1, D_2, D_3, D_{12}, D_{13}, D_{23}, and D_{123}, respectively. This implies that $y(p)$ is a random variable with mean y_0 and variance D. Each variance in (2.15) is called the partial variance corresponding to the subset of parameters; for example, D_{13} is the partial variance corresponding to the subset of parameters p_1 and p_3.

The Sobol global sensitivity indices are defined by

$$S_{i_1\cdots i_s} = \frac{D_{i_1\cdots i_s}}{D}, \quad i_1 < \cdots < i_s, \quad s = 1,2,3. \qquad (2.17)$$

For example, the first-order Sobol global sensitivity indices are defined by

$$S_i = \frac{D_i}{D}$$

and are used to compute the first-order contribution of the i^{th} parameter to the output variance. The second-order Sobol global sensitivity indices are defined by

$$S_{ij} = \frac{D_{ij}}{D}$$

and these measure the contribution to the variance from interaction between the i^{th} and j^{th} parameters.

The total sensitivity index of a parameter p_i is defined as the sum of all sensitivity indices for the subsets of parameters that include p_i. For example,

$$S_2^T = S_2 + S_{12} + S_{23} + S_{123}$$

is the total index for p_2. Given the definition of the sensitivity indices in (2.17), it is clear that the indices must sum to 1. For our particular example, we have

$$1 = S_1 + S_2 + S_3 + S_{12} + S_{13} + S_{23} + S_{123}. \tag{2.18}$$

Below, we consider a simple example to illustrate how to calculate the global sensitivity indices.

Example 2.3.1. We consider a specific case of the example discussed above. The model is given by

$$\frac{dy}{dt} = 6p_1^2 t + 4p_2p_3, \tag{2.19}$$

subject to

$$y(0) = 0,$$

where $p_1, p_2, p_3 \in I$. The solution of this initial value problem is given by

$$y(t;p) = 3p_1^2t^2 + 4p_2p_3t,$$

where $p = (p_1, p_2, p_3) \in I^3$.

Let $y(p) = y(t = 1; p) = 3p_1^2 + 4p_2p_3$. Using the formulae listed in (2.12), we now have that

$$y_0 = \int_0^1\int_0^1\int_0^1 (3p_1^2 + 4p_2p_3)dp_1dp_2dp_3 = 2,$$

$$y_1(p_1) = \int_0^1\int_0^1 (3p_1^2 + 4p_2p_3)dp_2dp_3 - 2 = 3p_1^2 - 1,$$

$$y_2(p_2) = \int_0^1\int_0^1 (3p_1^2 + 4p_2p_3)dp_1dp_3 - 2 = 2p_2 - 1,$$

$$y_3(p_3) = \int_0^1\int_0^1 (3p_1^2 + 4p_2p_3)dp_1dp_2 - 2 = 2p_3 - 1,$$

$$
\begin{aligned}
y_{12}(p_1,p_2) &= \int_0^1 (3p_1^2 + 4p_2p_3)dp_3 - 2 - y_1(p_1) - y_2(p_2) = 0,\\
y_{13}(p_1,p_3) &= \int_0^1 (3p_1^2 + 4p_2p_3)dp_2 - 2 - y_1(p_1) - y_3(p_3) = 0,\\
y_{23}(p_2,p_3) &= \int_0^1 (3p_1^2 + 4p_2p_3)dp_1 - 2 - y_2(p_2) - y_3(p_3)\\
&= 1 - 2p_2 - 2p_3 + 4p_2p_3,\\
y_{123}(p_1,p_2,p_3) &= y(p) - 2 - y_1(p_1) - y_2(p_2) - y_3(p_3)\\
&\qquad - y_{12}(p_1,p_2) - y_{13}(p_1,p_3) - y_{23}(p_2,p_3) = 0.
\end{aligned}
$$

Using the formulae listed in (2.15), we have

$$
\begin{aligned}
D &= \int_0^1\int_0^1\int_0^1 (3p_1^2 + 4p_2p_3)^2 dp_1dp_2dp_3 - 4 = 71/45,\\
D_1 &= \int_0^1 (3p_1^2 - 1)^2 dp_1 = 4/5,\\
D_2 &= \int_0^1 (2p_2 - 1)^2 dp_2 = 1/3,\\
D_3 &= \int_0^1 (2p_3 - 1)^2 dp_3 = 1/3,\\
D_{12} &= \int_0^1\int_0^1 0^2 dp_1dp_2 = 0,\\
D_{13} &= \int_0^1\int_0^1 0^2 dp_1dp_3 = 0,\\
D_{23} &= \int_0^1\int_0^1 (1 - 2p_2 - 2p_3 - 4p_2p_3)^2 dp_2dp_3 = 1/9,\\
D_{123} &= \int_0^1\int_0^1\int_0^1 0^2 dp_1dp_2dp_3 = 0.
\end{aligned}
$$

Therefore, the global sensitivity indices here are

$$\begin{aligned}
S_1 &= \frac{D_1}{D} = \frac{4}{5} \times \frac{45}{71} = \frac{36}{71},\\
S_2 &= \frac{D_2}{D} = \frac{1}{3} \times \frac{45}{71} = \frac{15}{71},\\
S_3 &= \frac{D_3}{D} = \frac{1}{5} \times \frac{45}{71} = \frac{15}{71},\\
S_{12} &= \frac{D_{12}}{D} = 0 \times \frac{45}{71} = 0,\\
S_{13} &= \frac{D_{13}}{D} = 0 \times \frac{45}{71} = 0,\\
S_{23} &= \frac{D_{23}}{D} = \frac{1}{9} \times \frac{45}{71} = \frac{5}{71},\\
S_{123} &= \frac{D_{123}}{D} = 0 \times \frac{45}{71} = 0.
\end{aligned}$$

We note that the model output is more sensitive to variations of the parameter p_1 than to those of the parameters p_2 and p_3. The interaction between the model parameters p_2 and p_3 modestly affects the model output. Finally, the interactions between the parameter p_1 and the other model parameters do not affect the model output since $S_{12} = S_{13} = S_{123} = 0$.

In summary, first-order sensitivity indices quantify the main effect of their corresponding parameter on the output. Second-order sensitivity indices are used to quantify the contribution of the interaction between two parameters to the output. Total sensitivity indices are used to compute the total contribution, including the main, second-order and higher-order effects, of a parameter to the output variance. The larger a sensitivity index is, the more influential the associated model parameter is. Although there are no distinct cutoff values defined for this type of analysis, the value of 0.05 is frequently accepted for distinguishing important from unimportant parameters [82]. In practice, it is difficult to directly calculate sensitivity indices, and they are usually computed numerically using computational packages, such as the SALib package for Python [83]. We discuss scientific packages for Python in the next section.

2.4 Some scientific Python packages

We now discuss the Python computational tools that are used in the remainder of the thesis. Python is an object-oriented, interpreted high-level programming language [84]. Python was created in the early 1990s by Guido van Rossum [85, 86, 84]. The readability of a Python program is usually high. In this section, we focus on the Python packages used for numerically integrating differential equations, optimising parameters, computing sensitivity indices, and plotting numerical results. These packages are NumPy, SciPy, SALib, and Matplotlib.

2.4.1 NumPy

NumPy is the fundamental package for scientific computing with Python. The ancestor of NumPy, `Numeric`, was created by Jim Hugunin, with contributions from several other developers [87]. In 2005, Travis Oliphant created NumPy by incorporating features of the competing `Numarray` into `Numeric`, with extensive modifications [88].

NumPy has the following features:

- a powerful N-dimensional array object,
- sophisticated (broadcasting) functions,
- tools for integrating C/C++ and Fortran code,
- useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases [89].

NumPy is open-source software and has many contributors. It is licensed under the BSD license, enabling reuse with few restrictions. A very complete manual by the principle author of NumPy, Travis Oliphant, is available for free [89, 90].

2.4.2 SciPy

SciPy is a collection of open-source Python-based software for mathematics, science, and engineering. In particular, it contains the following core packages [68, 87, 90]:

- NumPy, a base N-dimensional array package;
- SciPy library, a fundamental library for scientific computing;
- Matplotlib, a comprehensive 2D plotting package;
- IPython, an enhanced interactive console;
- Sympy, Symbolic mathematics; and
- Pandas, data structures and analysis.

SciPy contains modules for optimisation, linear algebra, integration, interpolation, special functions, fast Fourier transform, signal and image processing, ODE solvers, and other tasks common in science and engineering [68, 87]. SciPy was created in 2001 by Travis Oliphant, Eric Jones, and Pearu Peterson [87, 91]. The SciPy library contains a module of ODE solvers called `integrate`. One of these solvers is called `odeint`; it uses the `LSODA` program [92] from the FORTRAN library `odepack`, and it can be used to solve both stiff and non-stiff systems; see [93, 94, 95, 96].

The module `optimize` is contained in the SciPy library, and it is commonly used in scientific computing. This module provides several commonly used optimisation algorithms. The module contains [97]:

1. Unconstrained and constrained minimisation of multivariate scalar functions (minimise) using a wide range of algorithms, e.g. `Nelder-Mead simplex`, `SLSQP`, etc.
2. Global optimisation routines, e.g. `basinhopping`, `dual_annealing`, etc.
3. Least-squares minimisation (`least_squares`) and curve fitting algorithms (`curve_fit`).
4. Scalar univariate function minimisers (`minimize_scalar`) and root finders (`root_scalar`).
5. Multivariate equation systems solvers (`root`) using a variety of algorithms, e.g. `hybrid Powell`, `Levenberg-Marquardt`, etc.

In the next section, we discuss a Python package for implementing a sensitivity analysis.

2.4.3 SALib

SALib is an open-source Python library for implementing a sensitivity analysis. SALib was created by Jon Herman and Will Usher in 2016 [83]. To help understand how SALib works, we consider a model

$$y' = f(y, t, \mathbf{p}), \tag{2.20}$$

where $\mathbf{p} = (p_1, p_2, p_3)$ are the model parameters and the range of p_1, p_2, p_3 are $[9.0, 11.0]$, $[90.0, 110.0]$, and $[45.0, 55.0]$, respectively. Let $y(t, \mathbf{p})$ be the solution of the model obtained numerically by a Python solver. To calculate the Sobol global sensitivity indices of p_1, p_2, p_3 at $t = 1.0$, we present a Python program that consists of four main steps as follows:

- *Step 1.* Setup the problem for the sensitivity analysis.

```
problem = {
    'num_vars': 3, # number of variables
    'names':['p1', 'p2', 'p3'], # up to users
    'bounds': [[9.0, 11.0], [90.0, 110.0], [45.0, 55.0]]
}
```

- *Step 2.* Run the `sample` function to generate the model inputs. For a Sobol sensitivity analysis, the `sample` function is `saltelli`.

  ```
  param_values = saltelli.sample(problem, 1000)
  # 1000 is the number of samples.
  ```

- *Step 3.* For each sample $\mathbf{p}$ in Step 2, we evaluate the value of the model output $y(1.0, \mathbf{p})$, and save this output. Note that SALib does not do these calculations; for more details, see [98].

  ```
  Y = np.zeros([param_values.shape[0]])
  for j, X in enumerate(param_values):
      Y[j] = y(1.0, X)
      # y(1.0, X) must be a scalar value.
      # SALib does not calculate y(1.0, X) value.
  ```

- *Step 4.* Run the `analyze` function on the output file of Step 3 to compute the sensitivity indices.

  ```
  Si = sobol.analyze(problem, Y)
  ```

Note that this Python program needs some Python libraries to perform these four steps; for more details, see [98].

SALib contains several sensitivity analysis methods, such as Sobol [73, 99, 100], Morris [101, 77], and FAST [79, 78]. SALib is a useful and effective package for performing global sensitivity analyses and it is very easy to use.

2.4.4 Matplotlib

Matplotlib is a Python package that contains plotting tools that can produce publication-quality figures in a wide range of formats. Matplotlib is compatible with Python scripts and works well in the Python and IPython shells, the Jupyter notebook, and other web application servers [102].

Using Matplotlib we can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with a few lines of code [103]. For easy plotting, the `pyplot` module provides a MATLAB-like interface. The functionality of Matplotlib is also extended by several available toolkits, such as `basemap` [104] and `Mplot3d`; for more details, see [105].

The analysis cycle for a system of interest may be summarised as follows. First, using the law of mass action, we can write down ordinary differential equation models for systems of chemical reactions. These can be numerically solved using the `odeint` solver in the `integrate` module of the `SciPy` library. Values for the model parameters can be estimated by minimising the sum of squared residuals between the model outputs and experimental data. This may be implemented by calling one of the optimisation algorithms of the `minimize` submodule in the `optimize` module of the `SciPy` library. Once the values of the parameters are available, we can compute sensitivity indices for the parameters using the `SALib` package. Finally, `Matplotlib` can be used to produce publication-quality figures for the results obtained.

Chapter 3

Modelling hyaluronan degradation by *Streptococcus pneumoniae* hyaluronate lyase

The research presented in this chapter has been published in the *Journal of Mathematical Biosciences* [106].

Hyaluronic acid (Hyaluronan) is a linear, high molecular weight polysaccharide that forms an important component of the extracellular matrix. It is an excellent biomaterial, and it is increasingly being used in biotechnology, biomedical applications, and drug delivery. Polymer chains of hyaluronan occur in many different lengths in nature, and can be as large as multiples of ten thousand. Since the biological function of a hyaluronan chain often depends on its molecular weight, it is of value for applications to develop reliable quantitative descriptions of the degradation processes of hyaluronan. In particular, the development of such models should assist with the rational design of production processes to create polymer chains in a given molecular weight category for a specific application. In this chapter, we propose a new mathematical model for the degradation of hyaluronan by the enzyme *streptococcus pneumoniae* hyaluronate lyase. The model is based on a processive kinetic mechanism and consists of a coupled system of nonlinear ordinary differential equations for the species of interest. The model parameters are estimated using published experimental data, and good agreement between theory and experiment is found. Numerical experimentation and a Sobol global sensitivity analysis reveal that the key model parameters are the initial enzyme concentration and the rate constants for enzyme adsorption and catalysis.

The chapter is organised as follows. Motivation for this study is introduced in Section 3.1. In Section 3.2, we describe the formulation of the mathematical model. The computational methods used to analyse the model are described in Section 3.3, and the results and discussion are given in Section 3.4. We finish with conclusions in Section 3.5.

Nomenclature

$[A]$ -	the concentration of a species A; a function of time (units mg/ml)
D_0 -	the initial concentration of polymer chains of maximal degree (mg/ml)
D_i -	a free hyaluronic acid (HA) polymer molecule of length i disaccharide units
$\overline{D}_{i_1 i_2 \dots i_n}$ -	variation in model outputs w.r.t. changes in the model parameters $p_1, p_2, ..., p_n$ with $1 \leq n \leq 5$
$\overline{D}$ -	variation in model outputs w.r.t. changes in all of the model parameters
E_0 -	the initial concentration of enzyme (mg/ml)
E -	a molecule of the enzyme Streptococcus pneumoniae hyaluronate lyase (SpnHL)
$E \times D_i$ -	a HA-enzyme complex prior to the catalytic cleavage step
$E \circ D_i$ -	a HA-enzyme complex after the catalytic cleavage step, but prior to the translocation step
$E \diamond D_i$ -	a HA-enzyme complex after the translocation step
$J(\mathbf{p})$ -	sum of squares of residuals
k_{ads} -	adsorption rate of enzyme molecules to HA binding sites ($(mg/ml)^{-1}s^{-1}$)
k_{des} -	desorption rate of enzyme molecules from HA binding sites (s^{-1})
k_{clv} -	cleavage rate for the enzyme acting on a HA polymer chain (s^{-1})
k_{trans} -	rate constant for the translocation step (s^{-1})
k_{revtr} -	rate constant for reversing the translocation step (s^{-1})
$\mathbf{p}$ -	$(p_1, p_2, p_3, p_4, p_5) = (k_{ads}, k_{des}, k_{clv}, k_{trans}, k_{revtr})$, the model parameter set
$\mathbf{p}_0, \mathbf{p}^u, \mathbf{p}^l$ -	initial guesses, upper bounds, lower bounds for the parameters
R_i -	ratio of first-order sensitivity indices to total sensitivity indices
S_i, S_{ij} -	first-order, second-order sensitivity indices, respectively
$S_{i_1 i_2 \dots i_n}$ -	n^{th}-order sensitivity indices, $1 \leq n \leq 5$
S_i^{tot} -	total sensitivity indices
t -	time (s)
$\mathbf{y}$ -	vector of model outputs, that is, concentrations of various model species (mg/ml)
$z(t_i, \mathbf{p})$ -	model prediction for the total concentration of reducing ends at time t_i (mg/ml)
z_i -	experimentally measured value for the total concentration of reducing ends at time t_i (mg/ml)

3.1 Introduction

3.1.1 Background

Hyaluronan, also known as Hyaluronic acid, is a glycosaminoglycan that was originally discovered in the vitreous of bovine eyes by Karl Meyer and John Palmer in 1934 [107]. The glycosaminoglycans are a family of polysaccharides that are composed of repeating disaccharide units. For the case of hyaluronan, the repeating disaccharide unit is composed of the sugars D-glucuronic acid and N-acetylglucosamine connected by a β1,3 glycosidic bond; see Figure 3.1. These disaccharide units are in turn connected via β1,4 glycosidic bonds.

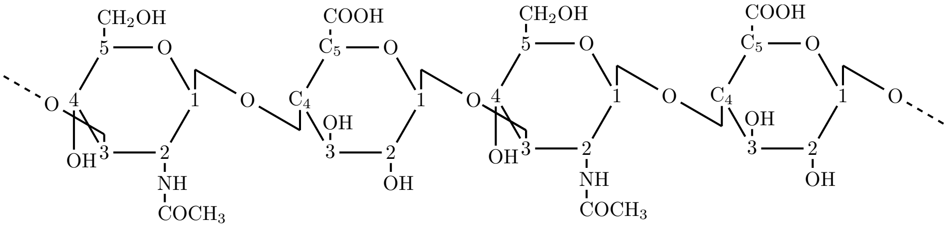


Figure 3.1: The chemical structure of hyaluronic acid. The central two units here form the repeating disaccharide, which is composed of the sugar D-glucuronic acid (left) and the sugar N-acetylglucosamine (right) connected by a β1,3 glycosidic bond. These disaccharide units are connected via β1,4 glycosidic bonds.

Hyaluronan can be found in most living organisms. In the human body, hyaluronan is present in the joints, the vitreous humor, the extracellular matrix, hair follicles, the gums, and the skin [108]. Hyaluronan is known to be involved in numerous biological processes, including inflammation, cell migration, and tumour development. Hyaluronan is a polyanion and its meshwork is known to be capable of sterically excluding other macromolecules. Hence, hyaluronan can play the role of a fence in protecting tissues from infection by bacteria [108]. Hyaluronan is also known to play the role of a shock absorber and lubricant in the body, as well as being involved in the transport of nutrients. Hyaluronan is an excellent biomaterial and is being increasingly used in biomedical applications, drug delivery, and tissue engineering [40, 109].

3.1.2 The function of hyaluronan depends on its molecular weight

In nature, hyaluronan polymer chains occur in many different lengths. The length of a hyaluronan polymer chain can be as large as multiples of ten thousand [40], and can have a molecular weight of the order of 10^7 Da. Intriguingly, it has been discovered that the biological function of a hyaluronan chain can be dependent on its chain length [108, 110]. Hyaluronan chains of 1000 saccharides or more have been shown to suppress angiogenesis [111], phagocytosis [112], hyaluronan synthesis [113], and the activity of the immune system [114]. On the other hand, shorter hyaluronan fragments of between 10 to 40 saccharides have been associated with CD44 cleavage (this refers to the removal via cleavage of the hyaluronan receptors CD44 from cell membranes) and the promotion of tumour cell migration [115]. Intermediate molecular weight hyaluronan has been shown to stimulate the expression of human β-defensin 2 (HBD2) in human keratinocytes [116].

Hence, it is of value for applications to develop reliable quantitative descriptions of the degradation processes of hyaluronan. In particular, the development of such models should assist with the rational design of production processes to generate polymer chains in a given molecular weight category for a specific application.

3.1.3 Hyaluronan degradation and hyaluronidases

The hyaluronidases are a family of enzymes that degrade hyaluronan [117, 118, 119, 120]. Hyaluronidases primarily fall into one of the followng three groups: eukaryotic hyaluronidases, invertebrate (leech) hyaluronidases, and bacterial hyaluronidases. In the eukaryotic category, humans have five hyaluronidases: HYAL1, HYAL2, HYAL3, HYAL4, and HYAL5 [118]. Almost all bacterial hyaluronidases degrade hyaluronan using an elimination mechanism, and so these enzymes are sometimes referred to as hyaluronan lyases.

Streptococcus pneumoniae is a pathogenic bacterium that is usually present in the upper respiratory tract of humans. It is responsible for many human diseases, including pneumonia, septicemia, otitis media, and bacterial meningitis [121]. *Streptococcus pneumoniae* hyaluronate lyase (SpnHL) [122, 123] is a surface enzyme of the *Streptococcus pneumoniae* bacterium that facilitates the invasion of the organism into animal tissue by degrading the connective tissue of the host, primarily hyaluronic acid [124].

3.1.4 The degradation mechanism of hyaluronan by SpnHL

The mechanism we outline here for the degradation of hyaluronan by SpnHL is based on work described in [7, 125, 126, 127, 39, 11]. The overall degradation mechanism is illustrated schematically in Figure 3.2; it may also be helpful to refer to Figure 3.5 here. The degradation process is processive and can be broken into five distinct steps as follows.

(i) *A random binding step.* An SpnHL enzyme randomly binds a hyaluronan chain; see [7] for structural details.

(ii) *A catalytic step.* The SpnHL enzyme cleaves a glycosidic β1,4 bond in the hyaluronan chain to create a truncated bound chain and an unreleased degradation product.

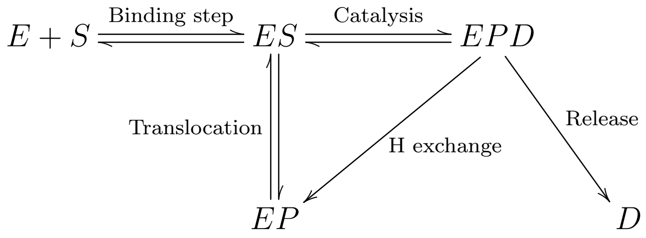


Figure 3.2: Adapted from [7]. Schematic of the overall processive degradation mechanism of hyaluronan by SpnHL. An SpnHL enzyme (E) initially randomly binds with a hyaluronan polymer chain substrate (S) to form an enzyme-polymer composite structure (ES). Following catalysis, the enzyme-polymer composite has been transformed to a truncated bound polymer subtrate with an unreleased product (EPD). The initial product (D) consists of an integer multiple of disaccharide units. The product (D) is then released to leave the truncated bound polymer substrate chain (EP). The enzyme then translocates along the truncated polymer substrate by one disaccharide unit toward the non-reducing end to recover the original bound state (ES). The process then repeats until the remaining polymer chain has been fully degraded, with all subsequent degradation products being single unsaturated disaccharide units.

(iii) *Hydrogen exchange.* During the catalytic step, the enzyme exchanges a hydrogen with the local water microenvironment; see [7] for structural details.

(iv) *Product release.* The degradation product is cleaved off from the hyaluronan chain to leave a bound truncated polymer. The degradation product for the first round of catalysis consists of an integer multiple of disaccharide units. All subsequent rounds of catalysis will produce a single unsaturated disaccharide degradation product (see Figure 3.3), as explained in the next step.

(v) *A translocation step.* The enzyme then translocates along the truncated polymer substrate by one disaccharide unit toward the non-reducing end to recover the original bound state. The process then repeats (move back up to step (ii)) until the remaining polymer chain has been fully degraded, with all subsequent degradation products being single unsaturated disaccharide units.

3.1.5 Some previous modelling studies

The literature for the mathematical modelling of hyaluronan degradation is not as well developed as that for cellulose [128, 129, 130], for example. Nevertheless, there are some mathematical studies that describe processive enzymatic mechanisms [131, 132, 133, 134]. The model described in [134] for cellulose degradation is particularly noteworthy. Like the model considered here,

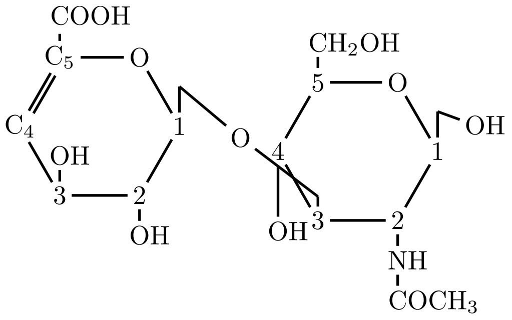


Figure 3.3: The chemical structure of an unsaturated disaccharide unit. Unsaturated discaccharide units are the end product of hyaluronan degradation by SpnHL.

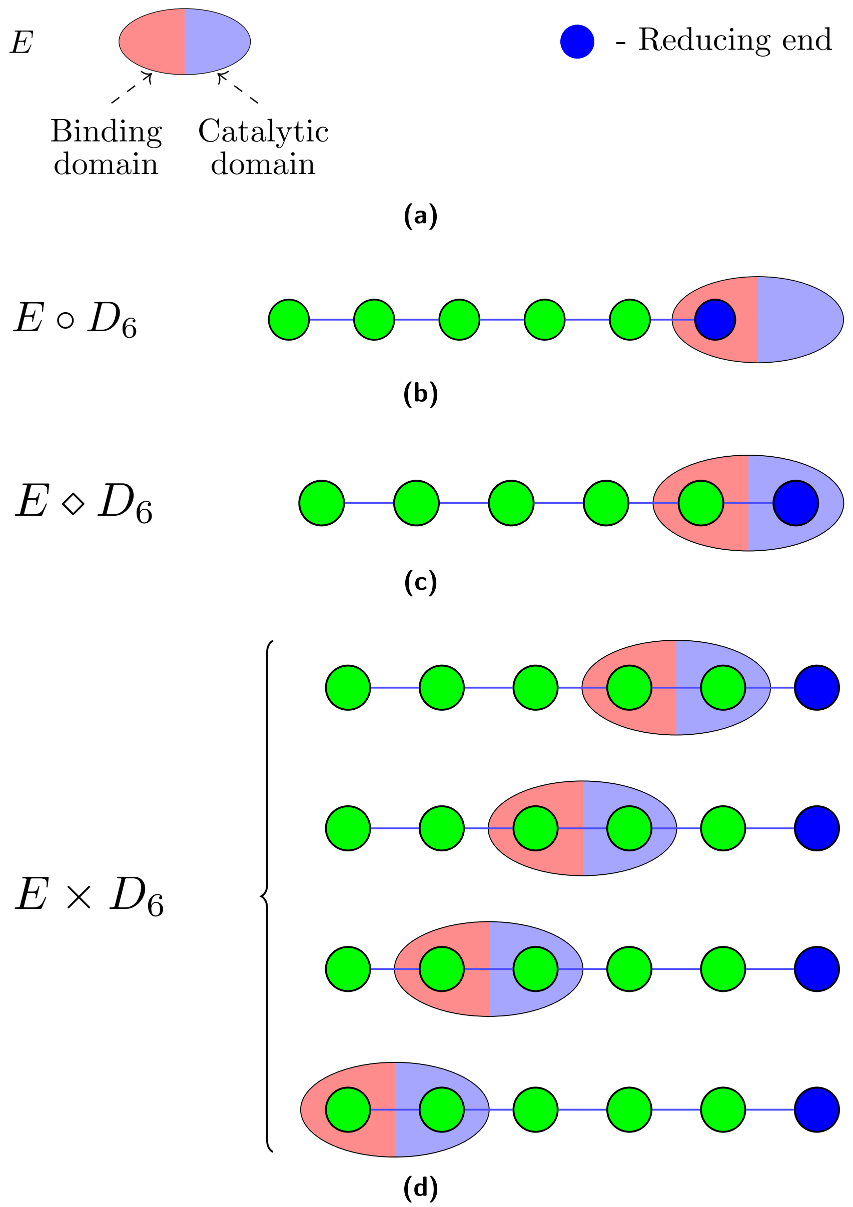
it is deterministic and explicitly incorporates a processive enzymatic mechanism. However, there are also significant differences with our modelling. For example, the model in [134] incorporates a steady-state assumption, whereas our model retains the full non-equilibrium equations. In the model we shall present, the initial binding step is random, so that there are numerous possibilities for the initial degradation product. Degradation then proceeds in the direction of the non-reducing end until the enzyme dissociates or the remaining fragment is degraded. In the model described in [134], there is no random binding step, and degradation proceeds for a *fixed* number n catalytic steps, where n corresponds to an experimentally measured mean processivity. In short, the equations we shall present capture more of the mechanistic detail of degradation by the lyase SpnHL [7], at the expense of a more complex model.

3.2 The mathematical model

In this section, we develop the mathematical model describing the degradation of hyaluronan by the hyaluronidase SpnHL. The model is detailed and tracks the evolution of the concentration of polymer chains of every possible degree in the mixture, from single disaccharides to chains of maximal length. We begin by listing our modelling assumptions.

3.2.1 Modelling assumptions

(a) It is assumed throughout that the degradation mixture of hyaluronan and SpnHL is well-stirred. This implies that diffusive effects in the degradation process can be neglected, and that the concentrations of the various species in the mixture can be described by functions of time only. This further implies that the evolution of the system can be modelled by a coupled system of nonlinear ordinary differential equations, and that a partial differential equations model is not required.



(a)

(b)

(c)

(d)

Figure 3.4: Diagrammatic representations for the three bound states of the enzyme and polymer that the mathematical model considers. (a) E: an enzyme molecule with a binding domain and a catalytic domain. (b) $E \circ D_6$: an enzyme molecule bound to the reducing end of a polymer fragment of degree six. (c) $E \diamond D_6$: an enzyme molecule bound to the disaccharide immediately to the left of the reducing end of the polymer fragment. During the processive phase of the degradation process, the enzyme cycles between an $E \circ D$ and an $E \diamond D$ state. (d) $E \times D_6$: The four possible states in which the polymer is bound to any of the four other binding sites of the polymer fragment. These states are required in the modelling to take account of the fact that the initial binding step is random.

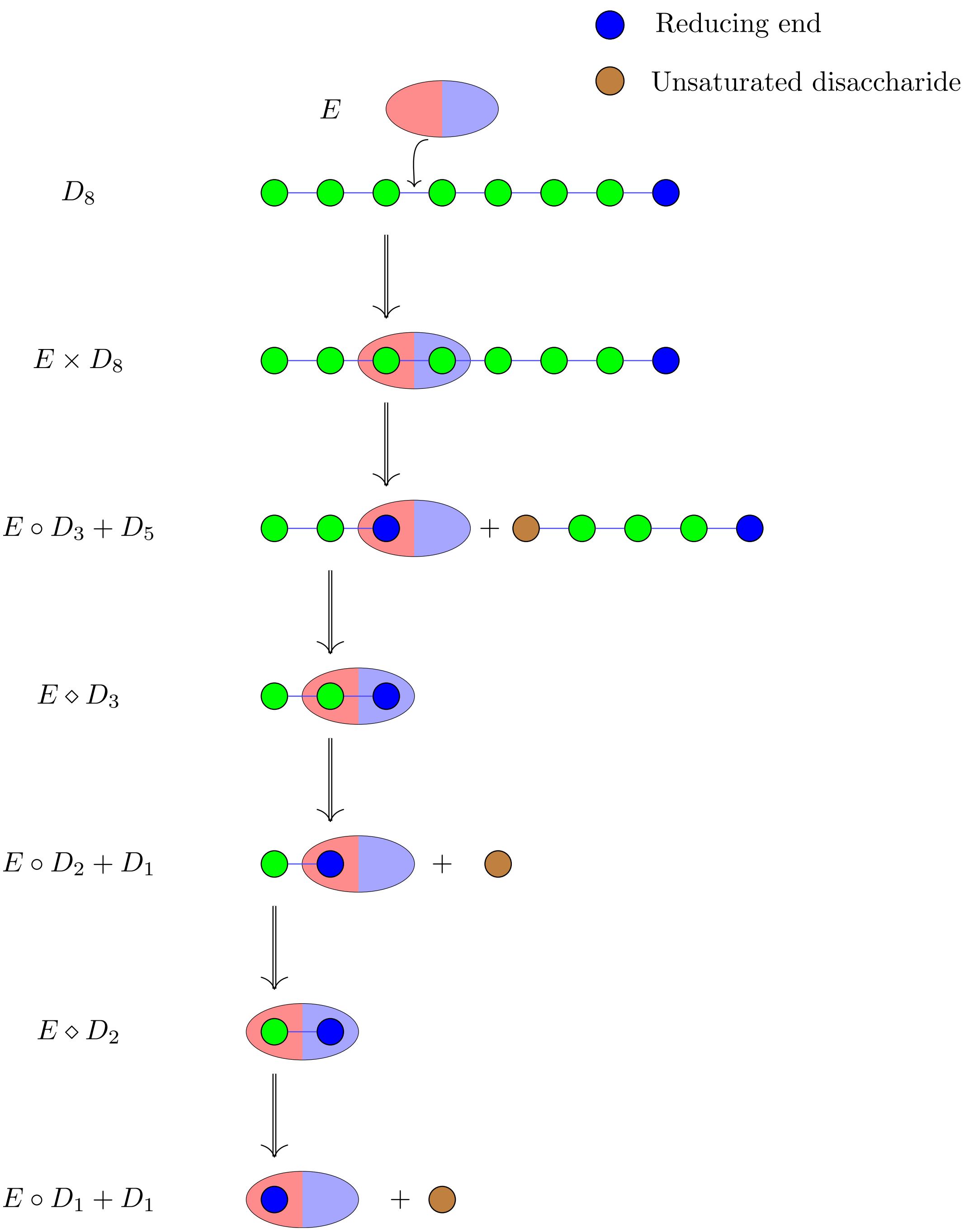


Figure 3.5: An example illustrating the degradation process of hyaluronan by SpnHL. An enzyme E binds to a polymer chain consisting of eight disaccharide units D_8 to create the bound state $E \times D_8$. In the particular example here, the enzyme happens to bind to the sixth disaccharide in the chain, counting from the reducing end. The enzyme cleaves a $\beta 1,4$ bond to create a bound state $E \circ D_3$ and a product D_5 as shown. From this point forward, the degradation is processive. The enzyme translocates one disaccharide unit along the polymer chain to arrive at the state $E \diamond D_3$. The enzyme then cleaves a $\beta 1,4$ bond to create a bound state $E \circ D_2$ and an unsaturated disaccharide product D_1. Following another translocation and degradation step, the polymer chain is degraded.

(b) We assume mass action kinetics throughout; this implies that the rate of a reaction is taken to be proportional to the product of the concentrations of the reactants. We emphasise here that more complex formulae, such as the Michaelis-Menten formula for the rate of product formation in an enzyme-catalysed reaction, are derivable from more fundamental mass action considerations under simplifying assumptions [18].

(c) We assume that the probability of an SpnHL enzyme binding with a polymer chain is proportional to the length of the polymer chain.

(d) The mechanism of Hyaluronan degradation by SpnHL is described in section 3.1.4 above. The mathematical model incorporates the following features: a random binding step, a combined catalytic and product release step, and a translocation step; see Figure 3.5. In the modelling, we allow for three distinct states for enzyme binding to the polymer, as shown in Figure 3.4. These are:

- $E \circ D_i$: the configuration in which the enzyme is bound to the reducing end of a polymer chain with i disaccharides (Figure 3.4 (a));
- $E \diamond D_i$: the configuration in which the enzyme is bound to the disaccharide unit immediately to the left of the reducing end (Figure 3.4 (b));
- $E \times D_i$: the $i-2$ configurations in which the enzyme is bound to any one of the other $i-2$ binding sites on the chain (Figure 3.4 (c)).

The character of the degradation process forces us to take account of these three states in the mathematical modelling, as we now explain. When the degradation process is in the processive phase, with the enzyme moving toward non-reducing end of the chain cleaving one dissacharide at a time, the system alternates between the configurations $E \circ D$ and $E \diamond D$. However, a third configuration, $E \times D$, is required to take account of the fact that the initial binding step is random, so that the initial degradation product can be larger than a single disaccharide. In Figure 3.5, we illustrate the degradation mechanism with a particular example.

(e) The binding and translocation steps are taken to be reversible. The combined catalytic and product release step is assumed to be irreversible.

(f) The model only allows for one enzyme to be bound to a polymer chain at a time.

$$E + D_1 \xrightarrow{k_{ads}} E \circ D_1 \xrightarrow{k_{trans}} E + D_1 \tag{3.1}$$

$$\begin{tikzcd}
& E \circ D_2 \arrow[d, shift left, "k_{trans}"] & & \\
E + D_2 \arrow[ur, "k_{ads}"] \arrow[dr, shift left, "k_{ads}"] & & & \\
& E \diamond D_2 \arrow[uu, shift left, "k_{revtr}"] \arrow[ul, shift left, "k_{des}"] \arrow[r, "k_{clv}"] & E \circ D_1 + D_1 \arrow[r, "k_{trans}"] & E + D_1.
\end{tikzcd} \tag{3.2}$$

$$\begin{tikzcd}
& E \circ D_i & & \\
E + D_i \arrow[ur, "k_{ads}"] \arrow[rr, shift left, "k_{ads}(i-2)"] \arrow[dr, shift left, "k_{ads}"] & & E \times D_i, \arrow[ll, shift left, "k_{des}"] & i \geq 3. \\
& E \diamond D_i \arrow[ul, shift left, "k_{des}"] & &
\end{tikzcd} \tag{3.3}$$

$$\begin{tikzcd}
& E \times D_i \arrow[r, "k_{clv}"] & E \circ D_{i-j} + D_j \arrow[d, shift left, "k_{trans}"] & \\
& E \circ D_{i-j-1} + D_1 \arrow[d, shift left, "k_{trans}"] & E \diamond D_{i-j} \arrow[u, shift left, "k_{revtr}"] \arrow[l, "k_{clv}"'] \arrow[r, "k_{des}"] & E + D_{i-j} \\
E + D_{i-j-1} & E \diamond D_{i-j-1} \arrow[u, shift left, "k_{revtr}"] \arrow[l, "k_{des}"'] \arrow[r, "k_{clv}"] & E \circ D_{i-j-2} + D_1 \arrow[d, shift left, dashed, "k_{trans}"] & \\
E + D_1 & E \circ D_1 + D_1 \arrow[l, "k_{trans}"'] & E \diamond D_2 \arrow[u, shift left, dashed, "k_{revtr}"] \arrow[l, "k_{clv}"'] \arrow[r, "k_{des}"] & E + D_2
\end{tikzcd} \tag{3.4}$$

Figure 3.6: The chemical reaction networks defining the mathematical model. Network (3.2) describes the degradation of a polymer fragment with just two disaccharides. Networks (3.3) and (3.4) need to be considered in combination, and describe the degradation of polymer fragments with three or more disaccharides. In (3.3) and (3.4), we have $i \geq 3$, $j < i$, and $j \geq 2$.

3.2.2 Construction of the governing ordinary differential equations

The complete set of governing equations for the model can be found in Appendix A. It is not necessary to discuss all of these equations here. However, we do briefly discuss two of them to illustrate how the governing equations are constructed. The chemical reaction networks for the model are displayed in Figure 3.6.

We begin by considering the equation for D_j where $3 \leq j < N$ and where D_N is a chain of maximal length. This is given by

$$\frac{d[D_j]}{dt} = \overbrace{-j\, k_{ads}[E]\,[D_j]}^{①} + \overbrace{k_{des}[E \diamond D_j]}^{②} + \overbrace{k_{des}[E \times D_j]}^{③} + \overbrace{k_{clv} \sum_{i=j+1}^{N} \frac{1}{i-2}[E \times D_i]}^{④}$$

where

① - this term accounts for the reduction in concentration of D_j due to enzyme binding. The j term is included here because the probability of enzyme binding is assumed to be proportional to the length of the polymer chain.

② - the increase in concentration of D_j due to enzyme unbinding from the complex $E \diamond D_j$.

③ - the increase in concentration of D_j due to enzyme unbinding from the complex $E \times D_j$.

④ - the increase in concentration of D_j due to the creation of degradation products D_j by the enzymatic cleavage of the complexes $E \times D_i$, where $i = j+1, j+2, ..., N$. The $1/(i-2)$ term is included here because there are $i-2$ possible configurations for each $E \times D_i$, and only one of these will produce the degradation product D_j.

It should be noted that $E \circ D_j$ terms do not appear in the above equation since it is assumed that enzyme may not unbind from the complexes $E \circ D_j$.

Next consider the equation for the enzyme E, given by

$$\frac{d[E]}{dt} = \overbrace{-k_{ads}[E]\left(\sum_{i=1}^{N} i[D_i]\right)}^{ⓐ} + \overbrace{k_{trans}[E \circ D_1]}^{ⓑ}$$
$$+ \overbrace{k_{des}\left(\sum_{i=2}^{N}[E \diamond D_i]\right)}^{ⓒ} + \overbrace{k_{des}\left(\sum_{i=3}^{N}[E \times D_i]\right)}^{ⓓ}$$

where

- ⓐ - this term accounts for the reduction in the free enzyme concentration due to enzyme binding with polymer chains. The i factor here is included because the probability of an enzyme binding to a polymer chain is assumed to be proportional to the length of the chain.

- ⓑ - the increase in the concentration of E due to enzyme unbinding from the complex $E \circ D_1$.

- ⓒ - the increase in the concentration of E due to enzyme unbinding from the complexes $E \diamond D_2$, $E \diamond D_3$, ..., $E \diamond D_N$.

- ⓓ - the increase in the concentration of E due to enzyme unbinding from the complexes $E \times D_3$, $E \times D_4$, ..., $E \times D_N$.

The remaining equations listed in Apppendix A are interpreted similarly. These equations are solved subject to the initial conditions

$$\begin{aligned}
&[E](t=0) = E_0, && \\
&[D_N](t=0) = D_0, && \\
&[D_i](t=0) = 0, && 1 \leq i \leq N-1, \\
&[E \times D_i](t=0) = 0, && 3 \leq i \leq N, \\
&[E \diamond D_i](t=0) = 0, && 2 \leq i \leq N. \\
&[E \circ D_i](t=0) = 0, && 1 \leq i \leq N,
\end{aligned}$$

where E_0, D_0 give the initial concentrations of enzyme and polymer chains of degree N, respectively. Notice here that we have chosen all of the polymer chains to have the same initial length N. Of course this is not realistic as we expect there to be some distribution of initial chain lengths in the mixture. In this context, our choice of N may be interpreted as the average initial length of the chains. However, we did carry out some numerical experiments for distributions of initial lengths for the polymer chains, and found that changing the character of the assumed initial conditions typically only had a weak effect on the values estimated for the key model parameters.

3.3 Computational methods

In this section, we describe the computational tools used to analyse the model equations. The software developed for this chapter was coded using the Python programming language [135].

3.3.1 Numerical solution of the ordinary differential equations

The system of differential equations was numerically integrated using the `odeint` solver in the module `integrate` of the SciPy library. SciPy [136] is an open source Python library that contains numerical routines for applications in science and engineering. The `odeint` solver [137] uses the `LSODA` program [138] from the FORTRAN library `odepack`, and it is capable of solving both stiff and non-stiff systems.

3.3.2 Parameter estimation

Experimental data

The parameter values were estimated with the aid of experimental data taken from the paper by Rapport *et al.* [11]. In their study, the type II strain D39R of *streptococcus pneumoniae* was used and sodium hyaluronate was sourced from umbilical cord. The incubation experiments were carried out at a temperature of 37° C and at a pH of 5.0; we emphasise here that the rate constants appearing in the mathematical model will in general depend on these two variables. Hence, it should be borne in mind that the parameter estimates we obtain are tied to the conditions of the incubation experiments. In [11], the progress of the degradation was quantified by measuring the concentration of reducing sugar in the incubation mixture. As the degradation proceeds, the concentration of reducing sugars increases. In the experiments, the initial concentrations of hyaluronate and enzyme were $D_0 = 10.0\ mg/ml$ and $E_0 = 1.0\ mg/ml$, respectively.

The method for parameter estimation

The experiments measure the total concentration of reducing ends, and so this is the quantity we work with in the estimation process. The total concentration of reducing ends is given in the model by

$$z(t;\mathbf{p}) = \sum_{i=1}^{N}[D_i](t) + \sum_{i=1}^{N}[E \circ D_i](t) + \sum_{i=2}^{N}[E \diamond D_i](t) + \sum_{i=3}^{N}[E \times D_i](t)$$

where $\mathbf{p}$ is the vector of model parameters. We denote the experimental data points by (t_i, z_i) for $1 \leq i \leq m$, where the z_i are the experimental measurements for the concentrations of the reducing ends, and the t_i are the corresponding times of measurement. We then form the following sum of squares of residuals

$$J(\mathbf{p}) = \sum_{i=1}^{m}(z(t_i, \mathbf{p}) - z_i)^2. \tag{3.5}$$

The vector of model parameters $\mathbf{p}$ is estimated by finding the vector $\mathbf{p}$ that minimises (3.5). This is a non-trivial minimisation task since the $z(t_i, \mathbf{p})$ here are determined by solving an initial value problem for a system of nonlinear ordinary differential equations (see Appendix A).

The minimisation was carried out using the routine `minimize` [139] in the module `optimize` of the SciPy library. The routine is provided with a set of initial guesses $\mathbf{p}_0$ for the parameters, as well as a set of parameter lower bounds $\mathbf{p}^l$ and upper bounds $\mathbf{p}^u$. For a given $\mathbf{p}$, $J(\mathbf{p})$ is calculated by calling the `odeint` solver (see Section 3.3.1) to calculate the $z(t_i, \mathbf{p})$. The routine used the Sequential Least SQuare Programming (SLSQP) [62, 140] method to perform the minimisation.

3.3.3 Global sensitivity analysis

A Sobol global sensitivity analysis was implemented to evaluate the importance of the various parameters appearing in the model [75, 73]. A Sobol analysis enables us to quantify how variations in the model parameters $\mathbf{p} = (p_i) = (k_{ads}, k_{des}, k_{clv}, k_{trans}, k_{revtr})$ affect the model output. This is achieved via the calculation of *sensitivity indices*. The model considered in this chapter has the structure $\mathbf{y} = \mathbf{f}(\mathbf{p}, t)$, where the inputs are the model parameters $\mathbf{p}$ and the time t, and the output $\mathbf{y}$ is a vector that gives the model predictions for the concentrations of the various species in the incubation mixture at time t. In the current chapter, we consider the effect on the output for the disaccharide concentration only, since disaccharides form the end product of the degradation process here.

For the parameter p_i, the associated first-order sensitivity index S_i is given by

$$S_i = \frac{\overline{D}_i}{\overline{D}},$$

where $\overline{D}_i$ is the variation of the model output with respect to changes in the parameter p_i, and $\overline{D}$ is the variation in the model output with respect to changes in all of the model parameters $\mathbf{p}$. For brevity, we do not explicitly display the formulae for $\overline{D}_i$ and $\overline{D}$ here; the details can be found in [75]. These first-order indices represent the effect of an individual parameter p_i on the output without interactions with the other parameters.

For the pair of parameters p_i and p_j $(i \neq j)$, the associated second-order sensitivity index S_{ij} is given by

$$S_{ij} = \frac{\overline{D}_{ij}}{\overline{D}},$$

where $\overline{D}_{ij}$ is the variation of the model output with respect to changes in the parameters p_i and p_j. This index measures the effect of the interaction between the parameters p_i and p_j on the model output. These ideas generalise in an obvious way for a set of parameters $p_i, p_j, ..., p_k$, where we define the sensitivity index

$$S_{ij...k} = \frac{\overline{D}_{ij...k}}{\overline{D}},$$

and where $\overline{D}_{ij...k}$ is the variation of the model output with respect to the parameters $p_i, p_j, ..., p_k$. The index $S_{ij...k}$ measures the effect of the interaction between the parameters $p_i, p_j, ..., p_k$ on the model output.

The total sensitivity index, S_i^{tot}, is the sum of all of the indices involving the parameter p_i, without repetition. It gives a measure for the total effect of the parameter p_i. Rather than display a rather opaque general formula ([75]) for S_i^{tot}, it is more instructive here to illustrate the idea with particular examples. If there are three parameters in total p_1, p_2, p_3, then

$$S_1^{tot} = S_1 + S_{12} + S_{13} + S_{123}.$$

For four parameters p_1, p_2, p_3, p_4, we have

$$S_2^{tot} = S_2 + S_{12} + S_{23} + S_{24} + S_{123} + S_{234} + S_{124} + S_{1234}.$$

Notice here that we have included S_{12}, but not $S_{21} = S_{12}$, and so on.

If it is found that the values S_i^{tot} and S_i are close, then the higher order indices are small, and this implies that interactions between the parameter p_i and the other model parameters do not significantly affect the model output.

For this chapter, the sensitivity analysis was implemented computationally using the Python package SALib [83, 141].

3.3.4 Polymer molecular weight averages

Knowledge of the molecular weight of polymer chains is useful in understanding the properties of a polymer, such as its mechanical strength, its solubility, or its chemical resistance. A polymeric material is generally a mixture of molecules differing in degree of polymerisation. As a result, some concepts of average molecular weights have been established to measure the molecular weight of a polymer material. Nowadays, number average molecular weight M_n and weight average molecular weight M_w are two important average molecular weights widely recognised and used; whereas two higher average molecular weights M_z and M_{z+1} tend to be used in measuring the motion of polymer molecules [142, 143].

Mathematical expressions for these average molecular weights are given by

$$M_n = \frac{\sum N_i M_i}{\sum N_i}, \tag{3.6}$$

$$M_w = \frac{\sum N_i M_i^2}{\sum N_i M_i}, \tag{3.7}$$

$$M_z = \frac{\sum N_i M_i^3}{\sum N_i M_i^2}, \tag{3.8}$$

$$M_{z+1} = \frac{\sum N_i M_i^4}{\sum N_i M_i^3}, \tag{3.9}$$

where M_i is the molecular weight of a chain and N_i is the number of chains of that molecular weight per ml. The formulae used to calculate the N_i and the

M_i are given by:

$$
\begin{aligned}
N_1 &= \frac{[D_1(t)]}{m} + \frac{[E \circ D_1(t)]}{m_E + m}, \\
N_2 &= \frac{[D_2(t)]}{2 \cdot m} + \frac{[E \circ D_2(t)] + [E \diamond D_2(t)]}{m_E + 2 \cdot m}, \\
N_i &= \frac{[D_i(t)]}{i \cdot m} + \frac{[E \circ D_i(t)] + [E \diamond D_i(t)] + [E \times D_i(t)]}{m_E + i \cdot m}, \quad i \geq 3, \\
M_i &= i \cdot m, \quad i \geq 1,
\end{aligned}
$$

where $m = 401.30$ Da is the molecular weight of a disaccharide unit, and $m_E = 8.3 \times 10^4$ Da is the molecular weight of the SpnHL enzyme.

From the expressions above, it follows that

$$M_n < M_w < M_z < M_{z+1}. \tag{3.10}$$

Additionally, the polydispersity index (PDI) is widely used to describe the width of a molecular weight distribution for a polymer, and is given by

$$PDI = \frac{M_w}{M_n}. \tag{3.11}$$

The bigger the polydispersity index is, the wider the molecular weight distribution is. A monodisperse polymer is a polymer with chains of equal length, such as a protein, so that $PDI = 1$.

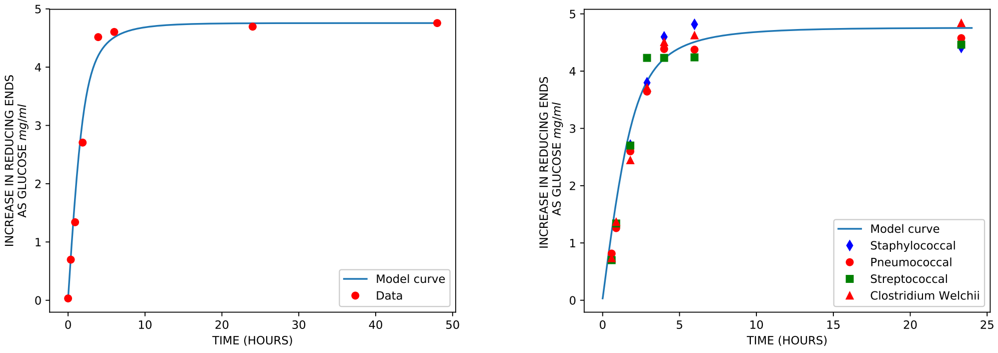
3.4 Results and discussion

3.4.1 Parameter estimation; comparison with experimental data

In Section 3.3.2, we give a discussion of experimental data and the method used to estimate the model parameters. We display the parameter values obtained in Table 3.1. These parameter values are appropriate for a temperature of 37°C and a pH of 5.0 [11], as dictated by the conditions of the experiments. Computational details such as the initial guesses and bounds for the parameters when calling the `minimize` routine can be found in the Appendix A. In the table, we display estimates for the parameters for a number of different values of N, where N gives the initial length of the polymer chains. Of course, under real experimental conditions, there will be a distribution of initial polymer chain lengths. Hence, we envisage that in the modelling presented here, N corresponds to the average of the initial polymer degree. In our numerical calculations, we found that the estimates for the parameters were rather insensitive to the values of N provided $N > 100$; in Table 3.1, compare the values for $N = 80$ and $N = 150$.

Table 3.1: Estimates for the parameter values obtaining by fitting with the experimental data of [11]. These values are appropriate for a temperature of 37°C and a pH of 5.0. Results are shown here for various values of N, the initial length of the polymer chains. The computational details can be found in the Appendix A.

N	k_{ads} (Molar^{-1}hr^{-1})	k_{des} (hr^{-1})	k_{clv} (hr^{-1})	k_{trans} (hr^{-1})	k_{revtr} (hr^{-1})
60	1.488×10^4	3.001×10^2	3.241×10^3	2.116×10^3	56.91
70	1.243×10^4	2.178×10^2	3.022×10^3	2.249×10^3	18.16
80	1.004×10^4	1.082×10^2	2.693×10^3	2.099×10^3	6.804
150	9.998×10^3	1.094×10^2	2.695×10^3	2.097×10^3	4.701



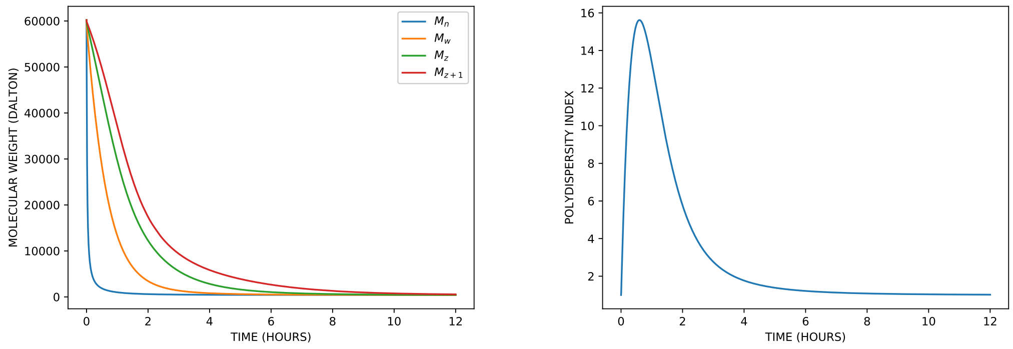
(a) **(b)**

Figure 3.7: Comparing theory and experiment for the degradation of hyaluronan by bacterial hyaluronidases. (a) A theoretical curve generated by the mathematical model developed here fitted to experimental data taken from [11]; see the main text. ●: Experimental data points for pneumococcal enzyme. —: The theoretical curve. (b) The same theoretical curve as in part (a) compared with (*not* fitted to) experimental data taken from [39]; see the main text. The symbols give the experimental data points for various enzymes. ♦: Staphylococcal enzyme. ●: Pneumococcal enzyme. ■: Streptococcal enzyme. ▲: Clostridium welchii enzyme. —: The theoretical curve.

In Figure 3.7 (a), we illustrate the results of the fitting process graphically. In this figure, we show a theoretical curve generated by the model fitted to experimental data taken from [11]. The parameters values that give the fitting can be found in Table 3.1 with $N = 150$. It is evident that the correspondence between theory and experiment is very good here, though this is hardly surprising here given that the parameter values were obtained by fitting the theoretical curve to this data. The units on the y axis of Figure 3.7 (a), *reducing ends as glucose mg/ml*, were those used in [11, 39]. A particular example serves to clarify their meaning - a value of 2 on this scale corresponds to that concentration of reducing ends in the incubation mixture that has the same reducing strength as a solution of 2 mg/ml of glucose. Unfortunately, due to a paucity of kinetic studies for SpnHL, there is no data we can directly compare the values in Table 1 with. However, there is quite an extensive literature on the degradation of cellulose, and some of the relevant parameters found for the cellulose models have the same order of magnitude as those displayed in Table 1; see, for example, [129, 130]. It should be said though that cellulose degradation is quite different to the system currently under consideration, so that such comparisons should be treated with caution.

In Figure 3.7 (b), we compare the same theoretical curve as that shown in (a) with experimental data taken from a different paper [39]. The conditions for the experiments in this paper are the same as those for [11]. However, in [39], experiments for four bacterial enzymes are considered, including *streptococcus pneumoniae*. We note again a good correspondence between theory and experiment, despite the fact that we are now *not* fitting the data to a theoretical curve. These results suggest that the degradation mechanism for *streptococcus pneumoniae* described here may be applicable to other bacterial enzymes as well.

While the correspondence between theory and experiment exhibited in Figure 7 (b) is gratifying and suggestive, more experimental data for *streptococcus pneumoniae* is required for model validation. In particular, data obtained under differential experimental conditions is required to properly assess the predictive capabilities of the model; for example, experiments with different initial concentrations for the enzyme or polymer.



(a) **(b)**

Figure 3.8: Plots of the average molecular weights and polydispersity index as functions of time. In this figure, the parameter values used to generate the numerical curves are given in Table 3.1 for $N = 150$. (a) The average molecular weights as given by the formulae (3.6), (3.7), (3.8), and (3.9). (b) The polydispersity index as a function of time given by the formula (3.11).

In Figure 3.8 (a), we plot the average molecular weights of the degrading polymer as functions of time using the formulae (3.6), (3.7), (3.8), and (3.9). It is immediately clear from the figure that the inequalities (3.10) are satisfied. The curves were generated using the parameter values in Table 3.1 for $N = 150$. In Figure 3.8 (b), we plot the corresponding polydispersity index as a function of time given by (3.11). Recall that the polydispersity index gives a measure of the broadness of the distribution of the molecular weight of a polymer sample. In Figure 3.8 (b), we see that the initial and final states are monodisperse, and that the maximum dispersity occurs after about one hour. Finally, Figure 3.9 shows concentrations of five groups of oligomers of disaccharides for the times $t = 0, 0.5$ and 1 hour. The behaviour exhibited is as expected with chain lengths decreasing in time.

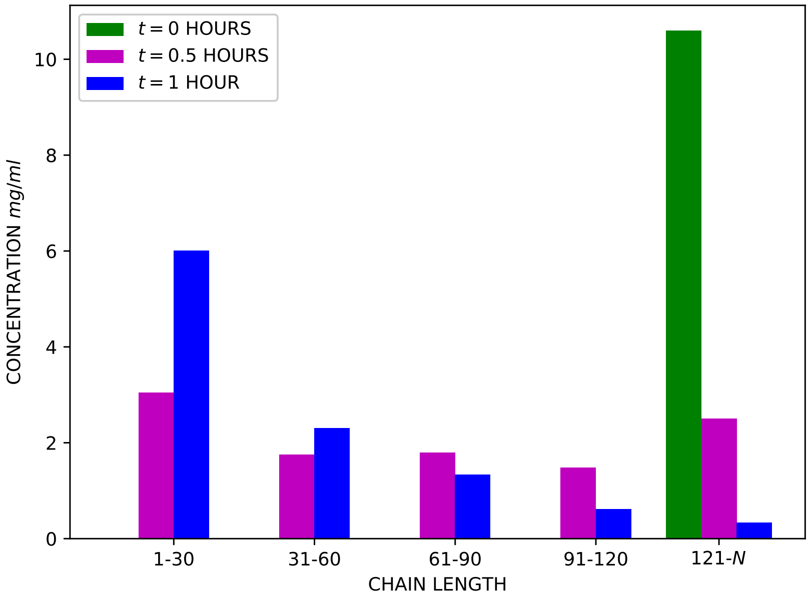


Figure 3.9: Concentrations of five groups of oligomers of disaccharides at times $t = 0, 0.5$, and 1 hour. The parameter values used to generate the numerical solutions are given in Table 3.1 for $N = 150$.

3.4.2 Global sensitivity analysis (GSA)

The Sobol global sensitivity analysis employed is discussed in Section 3.3.3, and was implemented using the SALib package with default parameters [83]. Our model has the structure $\mathbf{y} = \mathbf{f}(\mathbf{p}, t)$, where the inputs are the model parameters $\mathbf{p}$ and the time t, and the output $\mathbf{y}$ gives the model predictions for the concentrations of the species at time t. In the current study, we evaluate the effect on the output for the disaccharides only, and we calculate sensitivity indices for the times $t = 1$ hour, 2 hours, 3 hours, 4.5 hours, and 6 hours. Given the high computational cost of implementing a GSA here, we have chosen to use the modest values $N = 35$ and 40 for the initial lengths of the polymer chains. However, we shall see that the values of the sensitivity indices in our calculations are weakly dependent on the value chosen for N.

We display some of the results of the sensitivity analysis in Figure 3.10 and Figure 3.11. In all cases, the sensitivity indices for k_{revtr} are so small as to be difficult to see close to the x axis of the plots. Hence, we guess that the reverse

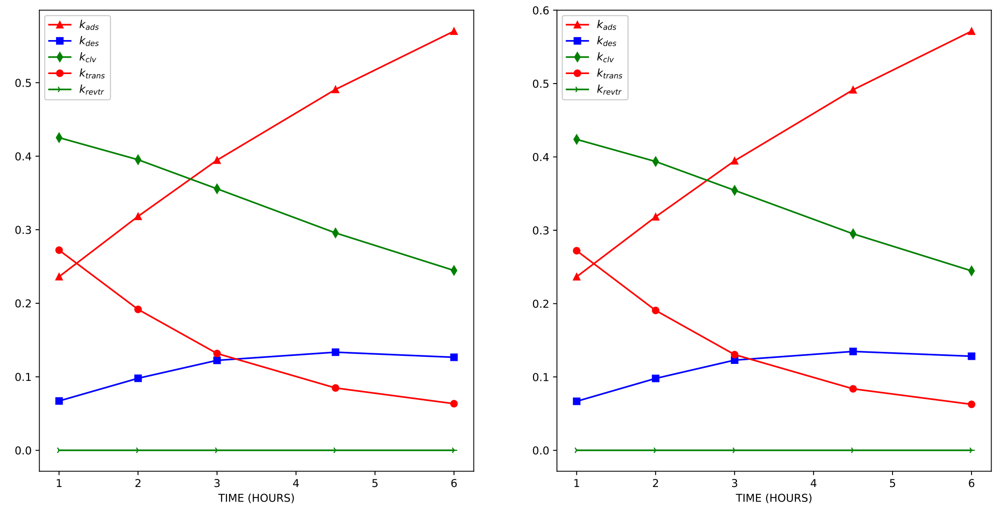


Figure 3.10: First order sensitivity indices (on the left) and total sensitivity indices (on the right) for the model parameters. Here the initial length of the polymer chains is $N = 35$. We see here that the first-order values and the total values are close so that parameter interactions do not significantly impact the model output here.

translocation step could be removed from the mathematical model developed here without significantly affecting the model output. However, the other four parameters k_{ads}, k_{des}, k_{clv}, k_{trans} do significantly affect model output, with the adsorption and catalytic parameters being the most important, and the desorption parameter being the least important. It is also evident that the sensitivity indices are functions of time, with the values of k_{ads} in particular rising significantly as the polymer degradation proceeds.

These results are supported in detail by the numerical curves for the disaccharide concentrations displayed in Figure 3.12. In this figure, the default parameter values used to generate the numerical curves are given in Table 3.1 for $N = 150$. Also, the initial concentration for the polymer is 10 mg/ml and the initial concentration for the enzyme is 1 mg/ml. The details of how the six plots in this figure are generated are explained in the figure caption. Inspecting the curves in these six plots, it appears from Figure 3.12 (f) that the output is most sensitive to variations in the initial enzyme concentration. However, variations in the initial enzyme concentration did not form part of our sensitivity analysis, and so we cannot compare it to our sensitivity results. Of the remaining five plots, Figure 3.12 (a) and Figure 3.12 (c) show the largest variation in behaviour, and these correspond to varying the values of k_{ads} and k_{clv}, respectively. This is consistent with the results of our sensitivity analysis where we have seen that k_{ads} and k_{clv} have the largest sensitivity indices. On the other hand, the curves in Figure 3.12 (e) show the least variation, and these correspond to variations in the values of k_{revtr}. This is again consistent with our sensitivity analysis where we have seen that k_{revtr} has by far the smallest sensitivity indices.

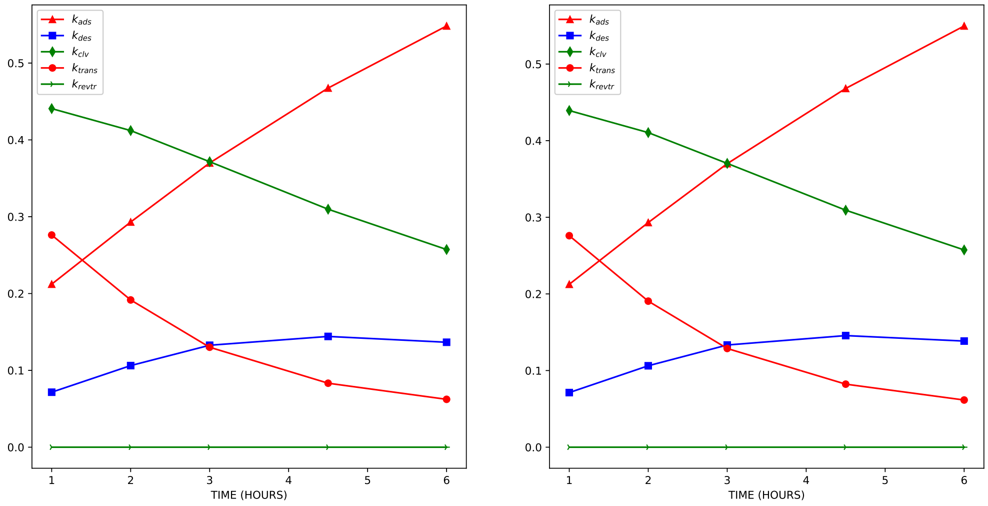


Figure 3.11: First order sensitivity indices (on the left) and total sensitivity indices (on the right) for the model parameters with $N = 40$. We see that the behaviour here is similar to that exhibited in Figure 3.10 which has $N = 35$, although there are perceptible differences in the values.

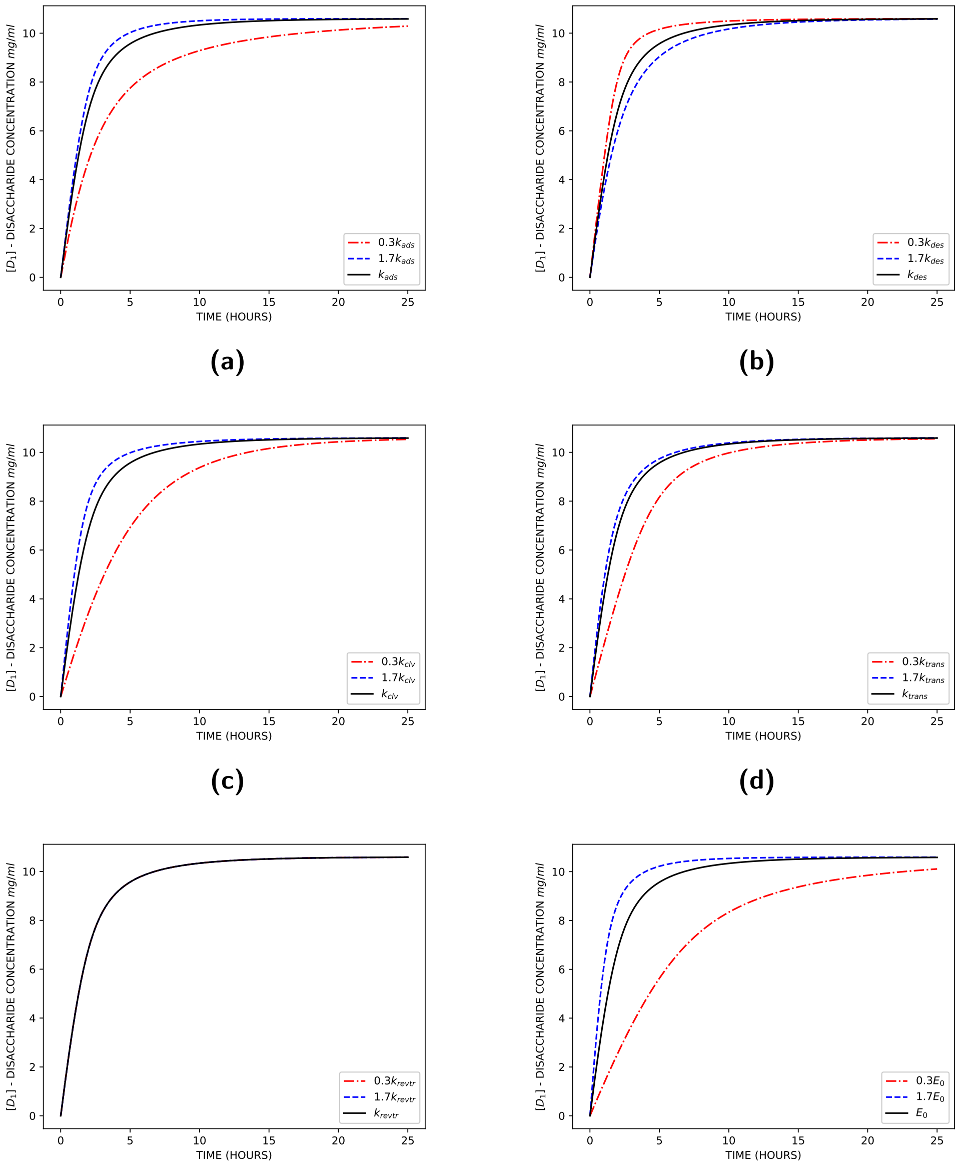
3.4.3 Other results

The mathematical model developed here is detailed and tracks the evolution in time of the concentration of all polymer fragments up to maximal degree. In Figure 3.13, we plot numerical solutions of the mathematical model for the concentrations of the polymer fragments D_2, D_3, D_4, and D_5. We note from these plots that the peak concentrations for the longer fragments occur at earlier times, as would be expected. In Figure 3.14, we plot concentrations of polymer fragments of various lengths, and for various initial concentrations of enzyme. We note again the strong dependence of the behaviour on the initial concentration of enzyme. This is of particular interest since the initial concentration of enzyme is a variable that is in principle under the control of the experimentalist/manufacturer.

Another quantity of interest here is the enzyme processivity. Following [134], we introduce the concept of a theoretical processivity, which is defined to be the mean number of sequential catalytic steps that would be performed on an infinitely long and uniform substrate strand. This number gives a measure for the actual processivity, the average number of catalytic steps for the real polymer substrate. Denoting the theoretical processivity by n_{theo}, and assuming that $n_{theo} \gg 1$, it is demonstrated in [134] that

$$n_{theo} \approx \frac{k_{des} + k_{clv}}{k_{des}},$$

where we recall that k_{des}, k_{clv} are the desorption and catalytic rate constants, respectively. For the current analysis, we found that for $N = 150$ and the



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 3.12: Numerical solutions of the mathematical model for disaccharide concentrations ($[D_1]$). These solutions support and illustrate the findings of the sensitivity analysis. In this figure, the default parameter values used to generate the numerical curves are given in Table 3.1 for $N = 150$. Also, the initial concentration for the polymer is 10 mg/ml and the initial concentration for the enzyme is 1 mg/ml. One parameter at a time is changed to generate the six plots in the figure. So, for example, to generate (a), all of the parameters are fixed at their default values, except for k_{ads}, which takes the values (default value)-70%[⋯], (default value) [—], (default value)+70%[- - -]. In (b), all of the parameters are fixed at their default values, except for k_{des}, which takes the values (default value)-70%[⋯], (default value) [—], (default value)+70%[- - -], and so on. We have (a) varying k_{ads}, (b) varying k_{des}, (c) varying k_{clv}, (d) varying k_{trans}, (e) varying k_{revtr}, and (f) varying E_0.

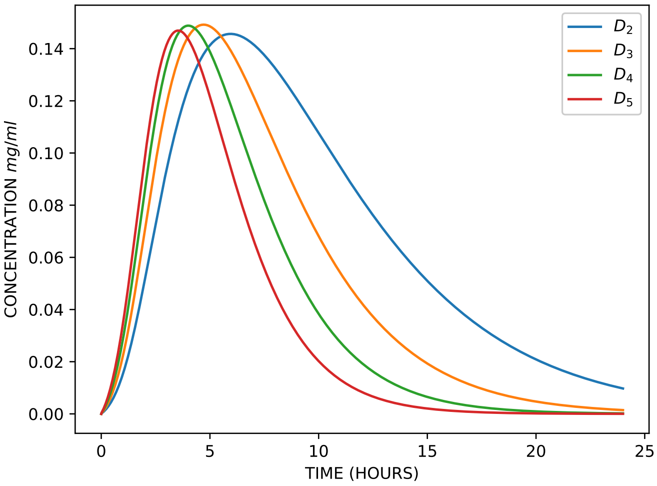


Figure 3.13: Numerical solutions of the mathematical model for the concentration of tetrasaccharides ($[D_2]$), hexasaccharides ($[D_3]$), octasaccharides ($[D_4]$), and oligomers of five disaccharides ($[D_5]$). The parameter values used to generate the curves are the same as the default values used for Figure 3.12.

experimental conditions descried in Section 3.2.1, we have $k_{des} = 1.094 \times 10^2$ hr^{-1}, $k_{clv} = 2.695 \times 10^3$ hr^{-1}, so that $n_{theo} \approx 26$.

3.5 Conclusions

Hyaluronan is a natural biopolymer that has numerous biomedical and industrial applications. The physiological function of hyaluronan chains can depend on their polymer degree. Hence, from the point of view of applications, the development of reliable mathematical models for the degradation of hyaluronan is clearly desirable. In this chapter, we develop the first detailed mathematical model for the degradation of hyaluronan by the bacterial hyaluronidase *streptococcus pneumoniae*. The model parameters values were estimated using available experimental data, and good agreement between theory and experiment was found. Furthermore, good agreement between the theory and hyaluronan degradation for other bacterial hyaluronidases was also seen, suggesting that the model may have wider applicability. Nevertheless, it should also be noted that additional experimental data is required for complete model validation. The model was further analyzed using numerical experimentation and a Sobol global sensitivity analysis, and it was found that the model output was most sensitive to the initial concentration of enzyme and the rate constants for enzyme adsorption and catalysis.

The model presented in the current study may be further refined. For example, it is known that hyaluronan in solution may adopt secondary and

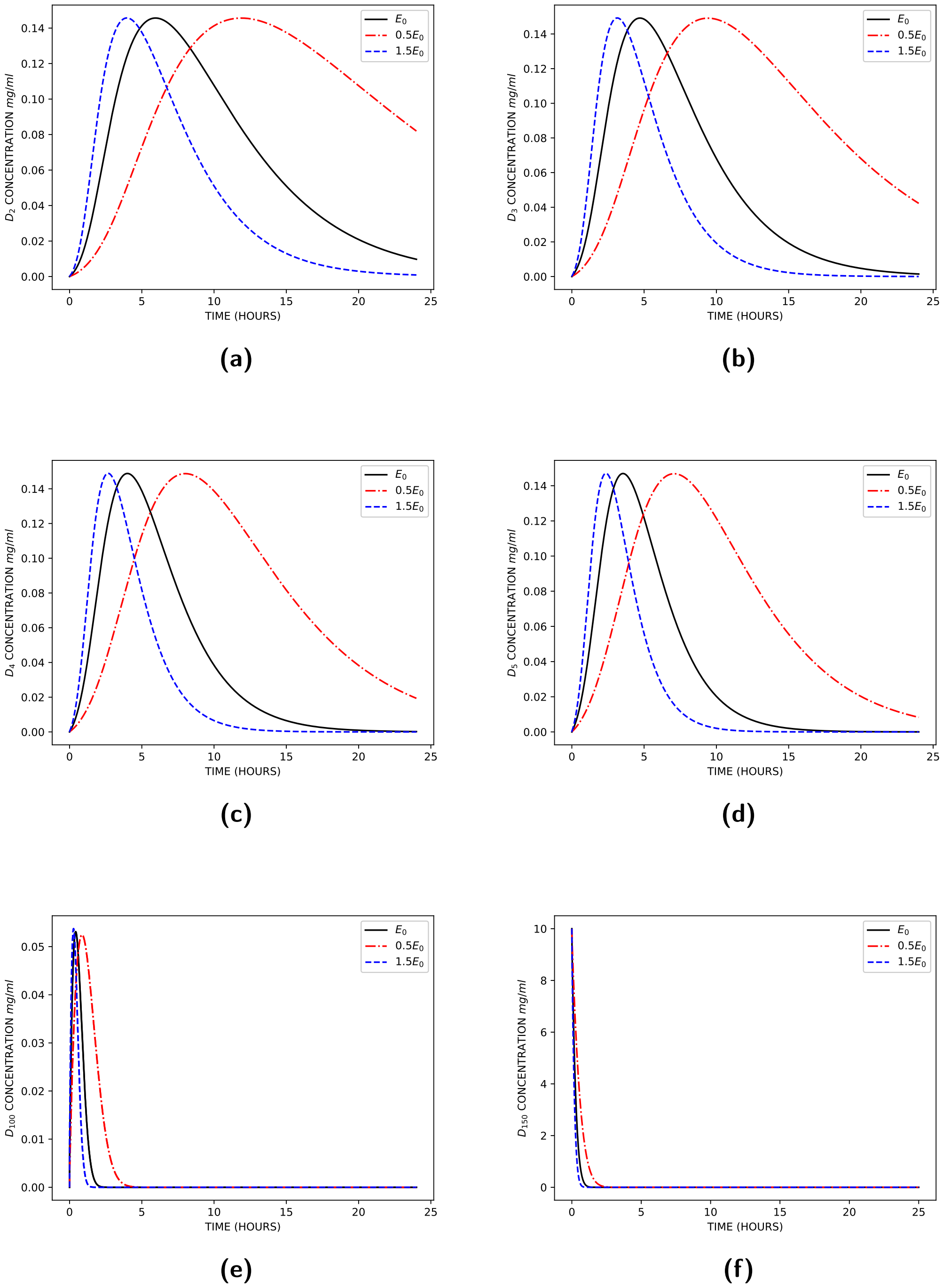


Figure 3.14: Numerical solutions of the mathematical model for the concentrations of the polymer fragments (a) D_2, (b) D_3, (c) D_4, (d) D_5, (e) D_{100}, and (f) D_{150}. The parameter values used here are the same as the default values used in Figure 3.12, except for the initial concentration of enzyme. Each of the six plots has three curves corresponding to three different initial concentrations for the enzyme, these being 0.5 mg/ml [- - -], 1.0 mg/ml [—], and 1.5 mg/ml [- - -].

tertiary structures [144, 145]. It is likely that these structures may affect the accessibility of some of the glycosidic bonds for the enzyme. Such effects have implications for the parameters in our modelling [146]. It is probable that some of the model parameters we are estimating are in fact effective parameters that implicitly incorporate effects not explicitly modelled. These issues could form the basis of future interesting studies that incorporate more of the mechanistic details of the degradation process. Another issue that requires further experimental and theoretical investigation is enzyme inhibition. In the current study, we assume that enzyme activity remains constant throughout the degradation process. However, many enzymatic degradation processes for polysaccharides are known to be subject to various inhibitory processes, and such effects may also play a role in the current context.

The analysis presented in the current study shows that the rate at which degradation proceeds is strongly dependent on the initial concentration of enzyme. The initial enzyme concentration is of particular interest since this quantity is potentially under the control of experimenters. Using numerical experimentation, quantitative insight into the relationship between the initial enzyme concentration and the rate of degradation has been established. This information should assist with the future design of degradation experiments for the SpnHL/hyaluronan system.

Chapter 4

Modelling the phosphorylation of glucose by human hexokinase I

The content of this chapter was submitted to the

Journal of Mathematical Biology

In this chapter, we have developed a comprehensive mathematical model to describe the phosphorylation of glucose by the enzyme hexokinase I. Glucose phosphorylation is the first step of the glycolytic pathway, and as such it is carefully regulated in cells. Hexokinase I phosphorylates glucose to produce glucose-6-phosphate, and the cell regulates the phosphorylation rate by inhibiting the action of this enzyme. The cell uses three inhibitory processes to regulate the enzyme: an allosteric inhibitory process, a competitive product inhibitory process, and a competitive inhibitory process. Surprisingly, the cellular regulation of hexokinase I is not yet fully resolved, and so in this chapter we have developed a detailed mathematical model to help unpick the behaviour. Numerical simulations of the model produce results that are consistent with the experimentally observed behaviour of hexokinase I. A global sensitivity analysis of the model was implemented to help identify the key mechanisms of hexokinase I regulation. The sensitivity analysis also enabled the development of a simpler model that produces output close to that of the full model. The computational software developed for this chapter has been made available in the Appendix B and online.

4.1 Introduction

Glucose is a major source of energy for most living organisms. Glucose glycolysis is a key pathway for the production of energy in a cell, and glycolytic intermediates form precursors for the biosynthesis of other key cellular constituents, such as glycogen, nucleotide sugars, and hyaluronan. The first step of glycolysis is the transformation of glucose into glucose-6-phosphate.

This is achieved via a phosphorylation that is catalysed by an enzyme called hexokinase. There are four isozymes of hexokinase found in mammalian tissue [147, 148], and these are usually referred to as hexokinase I, II, III, and IV (glucokinase). The molecular weights for hexokinase I, II, and III are all approximately 100 kDa. However, hexokinase IV is a smaller molecule, with a molecular weight of approximately 50 kDa [149].

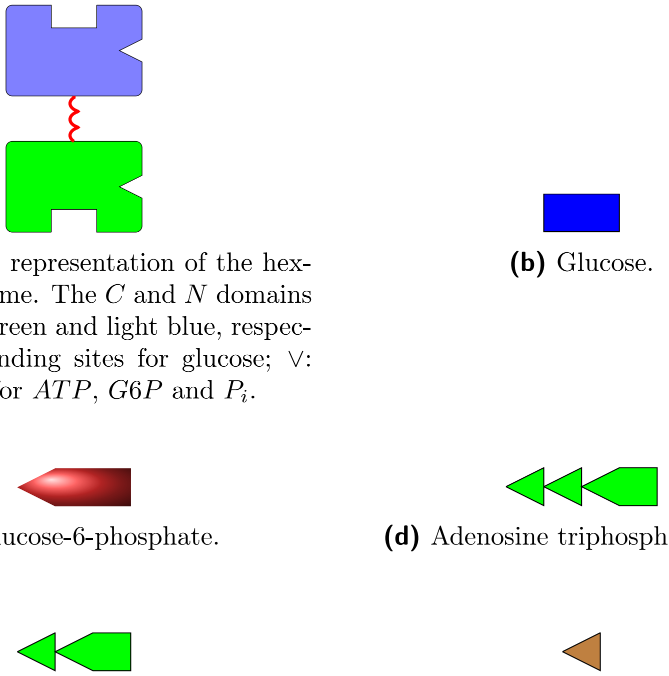
Previous studies have identified some of the functions and expression levels for the various hexokinase isoforms. Hexokinase I is present in all tissues, where it regulates the rate-limiting step of glycolysis; the mechanism of this regulation forms the topic of the current study. It is the predominant form present in brain cells and red blood cells [150, 151]. Hexokinase II is known to be highly expressed in skeletal muscle and adipose tissue [152, 153]. Hexokinase III is typically present at low levels in most tissues, with the highest levels being found in the lung, the kidney, and the liver [154, 155, 156]. Finally, glucokinase is primarily expressed in hepatocytes and pancreatic β cells [157, 158].

Hexokinase I and II can bind to the outer membrane of mitochondria, a process that has been associated with the prevention of cell death [159, 160]. Hexokinase III does not bind to mitochondria and exists predominantly in the cytoplasmic fraction, although there is evidence for Hexokinase III perinuclear binding [161]. Hexokinase III overexpression has been associated with a reduction in cell death [162]. Like hexokinase III, hexokinase IV (glucokinase) cannot bind to mitochondria and is localised in cytoplasm, where it plays a key role in the regulation of glucose homeostasis [163].

The product of glucose phosphorylation, glucose-6-phosphate ($G6P$), inhibits the activity of hexokinase I, II, and III (but not glucokinase) at physiological levels. Inorganic phosphate (P_i), however, antagonises the inhibition of hexokinase I by glucose-6-phosphate at low concentrations (few milimolar), and becomes an inhibitor of hexokinase I at high concentrations. In addition, inorganic phosphate inhibits hexokinases II and III at all concentrations [156, 162, 164]. Only the C terminal half of hexokinase I contains the catalytic sites, whereas the N terminal half does not [164, 165], but is involved in the P_i-antagonism of the product inhibition [165, 166]. In contrast, both the C and N terminal halves of hexokinase II are catalytically active and sensitive to $G6P$ levels [167, 168]. Furthermore, both hexokinase I and II have binding sites for ATP, glucose, $G6P$, and P_i in both N and C terminal halves [149, 165, 169]. Similar to hexokinase I, only the C-terminal half of hexokinase III is catalytically active [170, 171]. A detailed description of the kinetic mechanism for hexokinase I is given in the next section.

Many cellular factors can influence the phosphorylation of glucose by hexokinase I. In the current chapter, we construct a mathematical model that describes the cellular regulation of glucose phosphorylation. One of the principal aims of the modelling is to gain insight into the roles of $G6P$ and P_i in regulating the phosphorylation process. The model consists of a system of ordinary differential equations that tracks the evolution in time of the concentrations of various relevant species, including hexokinase enzyme, glucose, $G6P$, ATP, ADP, and P_i. We give schematic representations for each of these species in Figure 4.1.

Figure 4.1 (a) represents a single hexokinase I molecule, with blue being used for the N terminal domain and green for the C terminal domain. Each hexokinase I molecule possesses binding sites for glucose, ATP, $G6P$, and inorganic phosphate in the both C and N domains, even though the C domain only is catalytically active [164, 165, 149]. In Figure 4.1 (a), the binding sites for glucose on the C and N domains are depicted by the $\sqcup$ shape, and the binding sites for ATP, glucose-6-phosphate, and inorganic phosphate are represented by a $\vee$ cleft.



(a) Schematic representation of the hexokinase I enzyme. The C and N domains are coloured green and light blue, respectively. $\sqcup$: binding sites for glucose; $\vee$: binding sites for ATP, $G6P$ and P_i.

(b) Glucose.

(c) Glucose-6-phosphate.

(d) Adenosine triphosphate ATP.

(e) Adenosine diphosphate ADP.

(f) Inorganic phosphate P_i.

Figure 4.1: Schematics representations for hexokinase, glucose, glucose-6-phosphate, adenosine triphosphate (ATP), adenosine diphosphate (ADP), and inorganic phosphate P_i.

In 1969, *Ning et al.* [172] proposed a random Bi Bi kinetic mechanism for hexokinase I. This mechanism can be represented by the following set of chemical equations

$$\begin{tikzcd}
 & EA \arrow[dr, dash, "K_B"] & & & EC \arrow[dr, dash, "K'_A"] & \\
E \arrow[ur, dash, "K_D"] \arrow[dr, dash, "K_C"'] & & EAB \arrow[r, shift left, harpoon, "k_1"] & ECD \arrow[l, shift left, harpoon, "k_2"] \arrow[ur, dash, "K'_C"] \arrow[dr, dash, "K'_D"'] & & E, \\
 & EB \arrow[ur, dash, "K_A"'] & & & ED \arrow[ur, dash, "K'_B"'] &
\end{tikzcd} \tag{4.1}$$

where here E, A, B, C, D represent hexokinase enzyme, ATP, glucose, ADP, and $G6P$, respectively. Moreover, K_X, K'_X with $X = A, B, C, D$ are the dissociation constants for the four species. Finally, k_1 and k_2 are forward and backward rate constants, respectively, for the catalytic reaction. Some further discussion of this mechanism is given in Section 1.4 of Chapter 1. Many investigators have found experimental evidence in support of the Bi Bi mechanism for hexokinase I [149, 173, 174, 175, 176]. In the context of the current study, it forms a subset of a larger kinetic mechanism we develop for hexokinase I.

4.2 Mathematical model

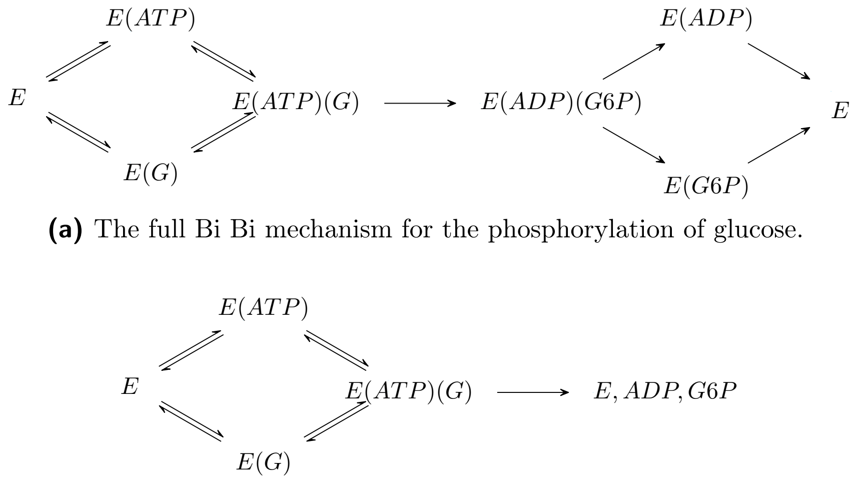
The mathematical model we develop for the phosphorylation of glucose will necessarily be large and complex since it will describe multiple binding sites, numerous species, almost 150 chemical reactions, and various inhibitory mechanisms. The phosphorylation mechanism of glucose by hexokinase I has already been briefly introduced in Section 4.1, and schematic representations of the relevant species arising are given in Figure 4.1. Recall that each hexokinase I molecule has two subunits, an N and a C terminal domain. Each subunit has its own binding site for glucose, and another binding site for ATP, P_i, and $G6P$ [149]. In Figure 4.3, we depict the eight possible configurations of the molecule where only one of binding sites is occupied.

4.2.1 The kinetic mechanism

The kinetic mechanism for the phosphorylation of glucose by hexokinase I may be summarized as follows.

1. *Binding sites.* Both the N and C domains have two binding sites; one for glucose and another for ATP, P_i, and $G6P$ [149, 177, 178, 179, 180, 166, 181, 182]; see Figure 4.3.

2. *Product formation.* The product here is $G6P$ and it is produced by the phosphorylation of glucose by hexokinase I. The phosphorylation is achieved via a Bi Bi mechanism [172], as represented in Figure 4.2 (a). However, in the current study, we consider the simplified process represented in Figure 4.2 (b). Here, for a $G6P$ molecule to be produced, an ATP molecule must be bound to its C domain site and a glucose molecule must be bound to its C domain site; see Figure 4.4.



(a) The full Bi Bi mechanism for the phosphorylation of glucose.

(b) The simplified Bi Bi mechanism modelled in the current study.

Figure 4.2: The Bi Bi mechanism for glucose phosphorylation.

3. *Product formation is regulated.* The phosphorylation of glucose is inhibited via the following three mechanisms.

   (a) Allosteric product inhibition. Binding of a molecule of $G6P$ to the N binding site makes a conformational change to the C domain binding site for ATP. This conformational change disables the binding of ATP to the C domain, resulting in the deactivation of the enzyme [179, 178]; see Figure 4.3 (g). This inhibition is mitigated by the presence of P_i and ATP which compete with $G6P$ for the N domain binding site.

   (b) Competitive product inhibition. The product $G6P$ competes with ATP for its C domain binding site, inhibiting product formation; see Figure 4.3 (h).

   (c) Competitive inhibition. P_i competes with ATP for the C domain binding site; see Figure 4.3 (f).

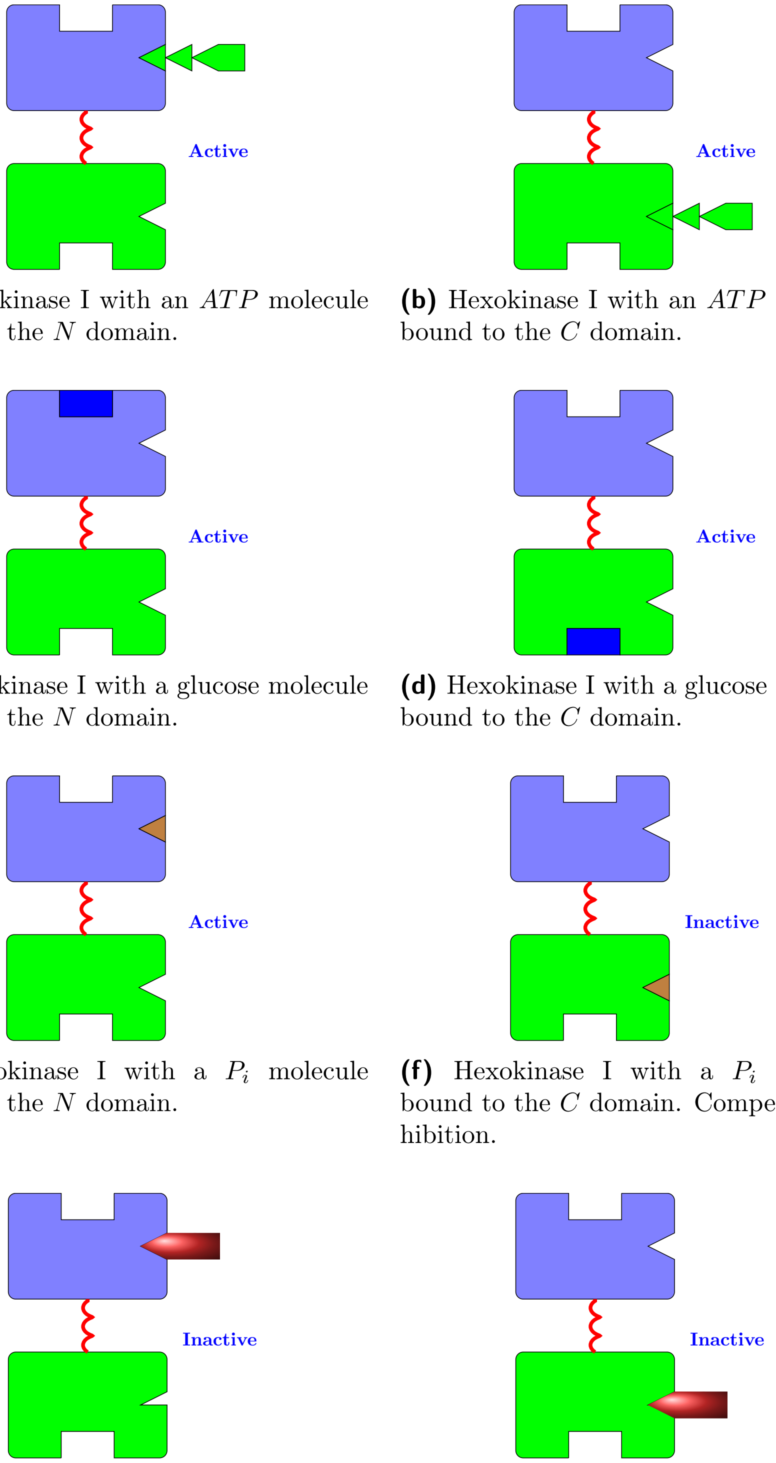
4. *Other details.* The following information concerning glucose phosphorylation is also available in the literature.

   (a) Only one molecule of $G6P$ can bind to an enzyme molecule at a time [178, 179].

   (b) The binding of P_i to the N domain binding site weakens the binding of $G6P$ to the C domain binding site (that is, it increases the dissociation constant) [180, 182].

   (c) The ATP binding sites of free or complexed enzyme are open, except for the case where a $G6P$ molecule is bound at the N binding site.

   (d) The high affinity binding site for $G6P$ is in the C domain, while the high affinity binding site for P_i is in the N domain [180].



(a) Hexokinase I with an ATP molecule bound to the N domain.

(b) Hexokinase I with an ATP molecule bound to the C domain.

(c) Hexokinase I with a glucose molecule bound to the N domain.

(d) Hexokinase I with a glucose molecule bound to the C domain.

(e) Hexokinase I with a P_i molecule bound to the N domain.

(f) Hexokinase I with a P_i molecule bound to the C domain. Competitive inhibition.

(g) Hexokinase I with an $G6P$ molecule bound to the N domain. Allosteric inhibition.

(h) Hexokinase I with an $G6P$ molecule bound to the C domain. Competitive product inhibition.

Figure 4.3: The eight possible configurations of a hexokinase molecule where only one of the binding sites is occupied. Figures (a),(b),(c),(d),(e) depict active states for the enzyme, while (f),(g),(h) depict inactive states.

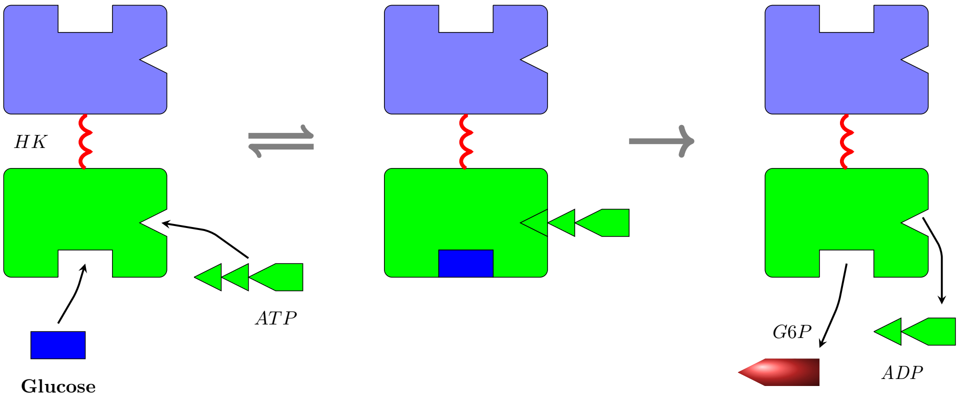


Figure 4.4: The mechanism for the phosphorylation of glucose by hexokinase I.

4.2.2 Modelling assumptions

(a) It is assumed throughout that the cellular mixture of hexokinase I, glucose, ATP, and P_i is well-stirred. This implies that diffusive effects in the phosphorylation process can be neglected, and that the concentrations of the various species in the mixture can be described by functions of time only. This further implies that the evolution of the system can be modelled by a coupled system of ordinary differential equations, and that a partial differential equation model is not required.

(b) We assume mass action kinetics throughout; this implies that the rate of a reaction is taken to be proportional to the product of the concentrations of the reactants. We emphasise here that more complex formulae, such as the Michaelis-Menten formula for the rate of product formation in an enzyme-catalysed reaction, are derivable from more fundamental mass action considerations under simplifying assumptions [183].

(c) We focus attention solely on the phosphorylation of glucose, and make no attempt to model in detail the evolution of the intracellular glucose concentration. Rather, we assume instead a constant initial concentration of glucose, and use the model to track its subsequent depletion as it is converted to $G6P$ via phosphorylation.

(d) The mechanism of the phosphorylation of glucose is assumed to be the simplified Bi Bi process represented in Figure 4.2b.

(e) The binding of one substrate does not affect the affinity of the binding sites for other substrates, except that the binding of P_i at the N domain reduces the affinity of the C binding site for $G6P$.

(f) The model allows only one molecule of $G6P$ to bind to an enzyme molecule at a time.

4.2.3 Model notation

We introduce the following model notation. We write

E :	a hexokinase I molecule,	0 :	a glucose molecule,	1 :	an ATP molecule,
2 :	a $G6P$ molecule,	3 :	a P_i molecule,	4 :	an ADP molecule.

We also add subscripts and superscripts to E, where a subscript denotes a molecule binding to the C domain of the enzyme, and a superscript denotes a molecule binding to its N domain. Hence, for example, we have

E_0 - A hexokinase I molecule with a glucose molecule bound to its C domain,

E^1 - A hexokinase I molecule with an ATP molecule bound to its N domain,

E_2^0 - A hexokinase I molecule with a $G6P$ molecule bound to its C domain and a glucose molecule bound to its N domain,

E_{03}^2 - A hexokinase I molecule with a P_i molecule and a glucose molecule bound to its C domain and a $G6P$ molecule bound to its N domain.

There are 59 such enzyme complexes in total, and so that there 65 species in all in the model; see Figure 4.5, 4.6 and the Appendix B. The concentration of a species X at time t will be denoted by $[X](t)$.

We introduce the following notation for the model rate constants.

k_0 : the catalytic constant (turnover rate) for hexokinase I,

k_1, k_3, k_5, k_7 : the forward rate constants for the binding of glucose, ATP, $G6P$, and P_i, respectively, to their N binding sites,

$k_{-1}, k_{-3}, k_{-5}, k_{-7}$: the reverse rate constants for the dissociation of glucose, ATP, $G6P$, and P_i, respectively, from their N binding sites,

k_2, k_4, k_6, k_8 : the forward rate constants for the binding of glucose, ATP, $G6P$, and P_i, respectively, to their C binding sites,

$k_{-2}, k_{-4}, k_{-6}, k_{-8}$: the reverse rate constants for the dissociation of glucose, ATP, $G6P$, and P_i, respectively, from their C binding sites,

k_9, k_{-9} : the forward and reverse rate constants for the binding/unbinding of $G6P$ to/from its C binding site, when the enzyme has a P_i molecule bound to its N binding site.

4.2.4 The chemical reactions

The system has numerous chemical reactions because of the large number of possible bound states for the enzyme, and a small selection of these are

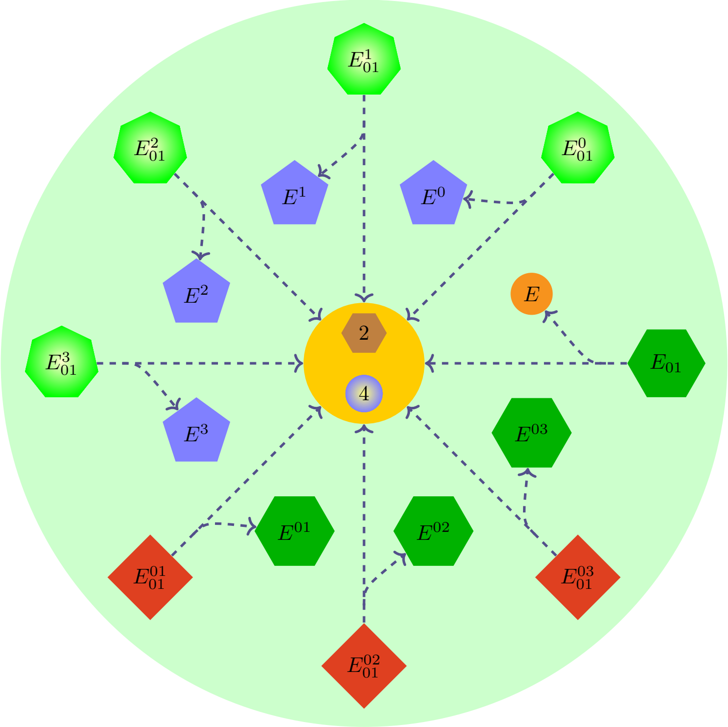


Figure 4.5: Diagram of chemical reactions producing product. The dashed lines represent irreversible reactions. There are eight such reaction types and each of them forms one $G6P$ and one ADP molecule (denoted by number 4), and either a free enzyme or an enzyme complex.

given by

$$\begin{array}{llll}
E+0 \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} E^0, & E+0 \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} E_0, & E+1 \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} E^1, & E+1 \underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}} E_1, \\
E+2 \underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}} E^2, & E+2 \underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}} E_2, & .... & ....
\end{array}$$

There are 147 reactions in all, and these are listed in detail in the Appendix B. In Figure 4.6, we schematically represent all of the chemical reactions producing an enzyme complex. Figure 4.5 shows the reactions that lead to the production of the product $G6P$.

4.2.5 Construction of the governing ordinary differential equations

The complete set of governing equations for the model can be found in the Appendix B, but it is not necessary to discuss all of these equations here. However, we do briefly discuss two of them to illustrate how the governing equations are constructed. The equations are developed based on the chemical reactions referred to in the previous subsection, and the law of mass action.

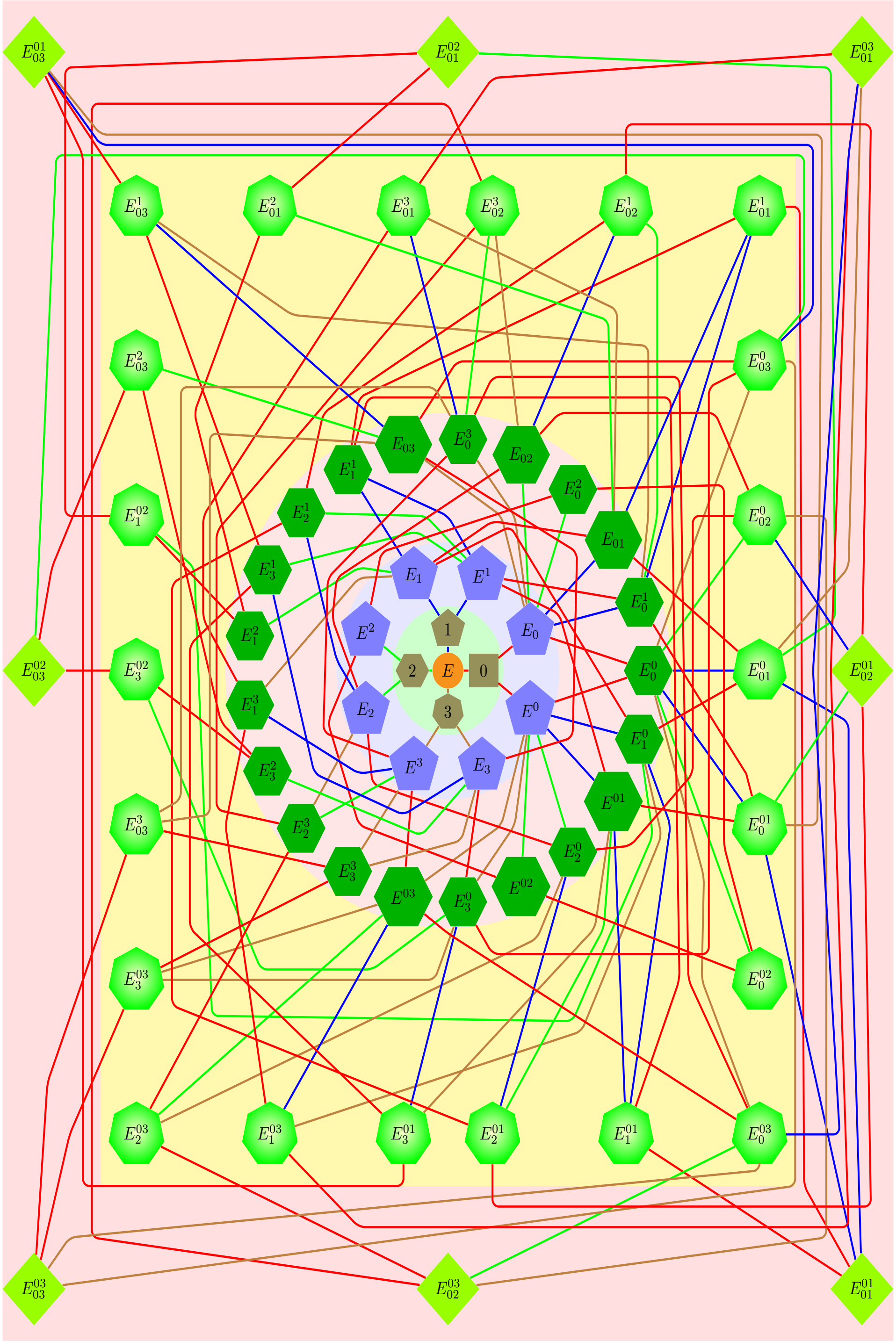


Figure 4.6: Diagram of chemical reactions forming all complexes in the mixture. E, 0, 1, 2, and 3 represent free enzyme, glucose, ATP, $G6P$, and P_i molecules, respectively. The red lines refer to reversible reactions involving glucose binding; the blue: ATP binding; the green: $G6P$ binding; the brown: P_i binding. Letters E with superscript(s) and/or subscript(s) denote complexes of enzyme, for instance, E_3^{01} is a complex of enzyme with one glucose and one ATP molecule bound at the N binding site, and one P_i molecule bound at the C binding site.

We begin by considering the equation for the product $G6P$, which is given by

$$
\begin{aligned}
\frac{d[2]}{dt} = {} & \overbrace{k_0([E_{01}] + [E_{01}^{0}] + [E_{01}^{1}] + [E_{01}^{2}] + [E_{01}^{3}] + [E_{01}^{01}] + [E_{01}^{02} + [E_{01}^{03}])}^{\textcircled{a}} \\
& + \overbrace{k_{-5}([E^{2}] + [E^{02}] + [E_{0}^{2}] + [E_{1}^{2}] + [E_{3}^{2}] + [E_{0}^{02}] + [E_{1}^{02}] + [E_{3}^{02}]}^{\textcircled{b}} \\
& \qquad + [E_{01}^{2}] + [E_{03}^{2}] + [E_{01}^{02}] + [E_{03}^{02}]) \\
& + \overbrace{k_{-6}([E_{2}] + [E_{02}] + [E_{2}^{0}] + [E_{2}^{1}] + [E_{02}^{0}] + [E_{02}^{1}] + [E_{2}^{01}] + [E_{02}^{01}])}^{\textcircled{c}} \\
& \qquad + \overbrace{k_{-9}([E_{2}^{3}] + [E_{2}^{03}] + [E_{02}^{3}] + [E_{02}^{03}])}^{\textcircled{d}} \\
& - \overbrace{[2]((k_5 + k_6)([E] + [E^{0}] + [E_{0}] + [E_{0}^{0}]) + k_5([E_{1}] + [E_{1}^{0}] + [E_{3}]}^{\textcircled{e}} \\
& \qquad + [E_{3}^{0}] + [E_{01}] + [E_{03}] + [E_{01}^{0}] + [E_{03}^{0}]) \\
& \qquad + k_9([E^{3}] + [E_{0}^{3}] + [E^{03}] + [E_{0}^{03}]) \\
& \qquad + k_6([E^{1}] + [E_{0}^{1}] + [E^{01}] + [E_{0}^{01}])),
\end{aligned}
\tag{4.2}
$$

where

- $\textcircled{a}$ - these terms accounts for the increase in the concentration of $G6P$ due to the creation of product by enzymatic reactions involving the complexes E_{01}, E_{01}^{0}, E_{01}^{k}, E_{01}^{0k}, where $k = 1, 2, 3$; see Figure 4.5.

- $\textcircled{b}$ - the increase in the concentration of $G6P$ due to $G6P$ unbinding from the enzyme complexes E^{2}, E^{02}, E_{k}^{2}, E_{k}^{02}, E_{0j}^{2}, E_{0j}^{02}, where $k = 0, 1, 3$ and $j = 1, 3$; see Figure 4.6.

- $\textcircled{c}$ - the increase in the concentration of $G6P$ due to $G6P$ unbinding from the enzyme complexes E_{2}, E_{02}, E_{2}^{01}, E_{02}^{01}, E_{2}^{k}, E_{02}^{k}, where $k = 0, 1$; see Figure 4.6.

- $\textcircled{d}$ - the increase in the concentration of $G6P$ due to $G6P$ unbinding from the complexes E_{2}^{3}, E_{2}^{03}, E_{02}^{3}, E_{02}^{03}; see Figure 4.6.

- $\textcircled{e}$ - the reduction in concentration of $G6P$ due to $G6P$ binding with the species E, E^{0}, E_{0}, E_{0}^{0}, E^{k}, E_{k}, E_{0}^{k}, E_{k}^{0}, E^{0k}, E_{0k}, E_{0}^{0k}, E_{0k}^{0}, where $k = 1, 3$; see Figure 4.6.

Next consider the equation for the enzyme concentration, given by

$$\begin{aligned}\frac{d[E]}{dt} &= \overbrace{k_0[E_{01}]}^{\textcircled{1}} \\ &\quad + \overbrace{k_{-1}[E^0] + k_{-2}[E_0] + k_{-3}[E^1] + k_{-4}[E_1] + k_{-5}[E^2]}^{\textcircled{2}} \\ &\qquad\qquad + k_{-6}[E_2] + k_{-7}[E^3] + k_{-8}[E_3] \\ &\quad - \overbrace{[E]((k_1 + k_2)[0] + (k_3 + k_4)[1] + (k_5 + k_6)[2] + (k_7 + k_8)[3])}^{\textcircled{3}}, \end{aligned} \tag{4.3}$$

where

① - this term accounts for the increase in concentration of enzyme due to the recovery of enzyme after the catalytic step has been completed.

② - this gives the increase in the concentration of the free enzyme due to the unbinding from the complexes E^k, E_k, where $k = 0, 1, 2, 3$.

③ - this gives the reduction in the concentration of enzyme due to enzyme binding.

4.2.6 Initial conditions

The equations described in the previous subsection are solved subject to the initial conditions

$$\begin{array}{lll}[E](t = 0) = E_0, & [0](t = 0) = G_0, & [1](t = 0) = ATP_0, \\ [2](t = 0) = 0, & [3](t = 0) = P_{i0}, & [4](t = 0) = 0,\end{array}$$

where E_0, G_0, ATP_0, P_{i0} give the initial constant concentrations of enzyme, glucose, ATP, and P_i, respectively. The initial concentrations for all of the enzyme complexes were taken to be zero.

4.2.7 Computational methods

In this section, we describe the computational tools used to analyse the model equations. The software developed for this chapter was coded using the Python programming language [84].

Numerical method for solving the ordinary differential equations

The system of differential equations was numerically integrated using the `odeint` solver in the module `integrate` of the SciPy library. SciPy [68] is an open source Python library that contains numerical routines for applications in science and engineering. The `odeint` solver [184] uses the `LSODA` program [92] from the FORTRAN library `odepack`, and it is capable of solving both stiff and non-stiff systems.

Model parameter values

Table 4.1 shows some of the model parameter values, together with their literature sources. We note that the dissociation constant for P_i at its C binding site has been taken to be ten times larger than its value at the N site [185]. This implies that the higher affinity binding site for P_i is in the N domain, with much weaker binding to the C site. We recall that the enzyme is active when P_i is bound to its N site (Figure 4.3 (e)) but inactive when bound to the C site (Figure 4.3 (f)). Hence, for low concentrations of P_i (few milimolar), the higher affinity N site dominates and the inhibition of $G6P$ is antagonised. However, for higher P_i concentrations, P_i binding to the C site is significant and enzyme activity is inhibited. This behaviour matches experimental findings [149, 156, 182].

Table 4.1: Some model parameter values and their literature sources.

Substrate	Domain	$K_m(\mu M)$	$K_d(\mu M)$	$k_{cat}(k_0)s^{-1}$	Ref.
Hexokinase I				63	[180]
Glucose	C	53			[180]
ATP	C	700			[180]
$G6P$	N		710		[180]
	C		54		[180]
P_i	N		22		[182]
	C		220		[185]

The Lambda (Λ) and Omega (Ω) methods for approximating kinetic rate constants are discussied in the paper [186]. Rate constants for the current model were estimated using these methods with $\Lambda = 100$, $\Omega = 1.0$ and the data displayed in Table 4.1; see Table 4.2. The Michaelis-Menten constants for the C domain binding sites of glucose and ATP are known. The corresponding values for the N domain are unknown, and so in the absence of other information, are taken here to be the same as their C domain values.

Table 4.3 displays typical intracellular concentrations for Hexokinase I and some other model species. These values informed the choice of initial conditions for the numerical solutions.

Global sensitivity analysis

A Sobol global sensitivity analysis was implemented to evaluate the importance of the various parameters appearing in the model [73, 75]. The model

Table 4.2: Values for the model rate constants.

Parameter	Description	Value	Unit
k_0	Catalytic constant	63	s^{-1}
k_1	Forward rate const. for glucose to N site	$1.18868{\times}10^5$	$mM^{-1}s^{-1}$
k_{-1}	Reverse rate const. for glucose from N site	$6.237{\times}10^3$	s^{-1}
k_2	Forward rate const. for glucose to C site	$1.18868{\times}10^5$	$mM^{-1}s^{-1}$
k_{-2}	Reverse rate const. for glucose from C site	$6.237{\times}10^3$	s^{-1}
k_3	Forward rate const. for ATP to N site	$9.0{\times}10^3$	$mM^{-1}s^{-1}$
k_{-3}	Reverse rate const. for ATP from N site	$6.237{\times}10^3$	s^{-1}
k_4	Forward rate const. for ATP to C site	$9.0{\times}10^3$	$mM^{-1}s^{-1}$
k_{-4}	Reverse rate const. for ATP from C site	$6.237{\times}10^3$	s^{-1}
k_5	Forward rate const. for $G6P$ to N site	$9.0{\times}10^3$	$mM^{-1}s^{-1}$
k_{-5}	Reverse rate const. for $G6P$ from N site	$6.390{\times}10^3$	s^{-1}
k_6	Forward rate const. for $G6P$ to C site	$9.0{\times}10^3$	$mM^{-1}s^{-1}$
k_{-6}	Reverse rate const. for $G6P$ from C site	$4.86{\times}10^2$	s^{-1}
k_7	Forward rate const. for P_i to N site	$9.0{\times}10^3$	$mM^{-1}s^{-1}$
k_{-7}	Reverse rate const. for P_i from N site	$1.98{\times}10^2$	s^{-1}
k_8	Forward rate const. for P_i to C site	$9.0{\times}10^3$	$mM^{-1}s^{-1}$
k_{-8}	Reverse rate const. for P_i from C site	$1.980{\times}10^3$	s^{-1}
k_9	Forward rate const. for $G6P$ to E^3, E^{03}	$9.0{\times}10^2$	$mM^{-1}s^{-1}$
k_{-9}	Reverse rate const. for $G6P$ from E^3, E^{03}	90	s^{-1}

Table 4.3: Intracellular concentrations of Hexokinase I and some metabolites.

Substrate	Concentration (mM)	Cell type	Ref.
Hexokinase I	$6.65{\times}10^{-2}$	General cells	[187]
Glucose	2.5	Adipose cells	[188]
ATP	3.0	Brain cells	[189]
$G6P$	0 - 3.0		
P_i	0.0 -15.0	Brain cells	[189, 182]

considered in this chapter has the structure $\mathbf{y} = \mathbf{f}(\mathbf{p}, t)$, where the inputs are the model parameters $\mathbf{p}$ and time t, and the output $\mathbf{y}$ is a vector that gives the model predictions for the concentrations of the various species at time t. A Sobol sensitivity analysis enables us to quantify how variations in the model parameters $\mathbf{p} = (p_i)$, affect the model output $\mathbf{y}$. This is achieved via the calculation of *sensitivity indices.* A detailed explanation of sensitivity indices can be found in Section 3.3.3 of Chapter 3.

The sensitivity analysis in the current study was implemented computationally using the Python package SALib [83, 141].

4.3 Results and discussion

4.3.1 Numerical results

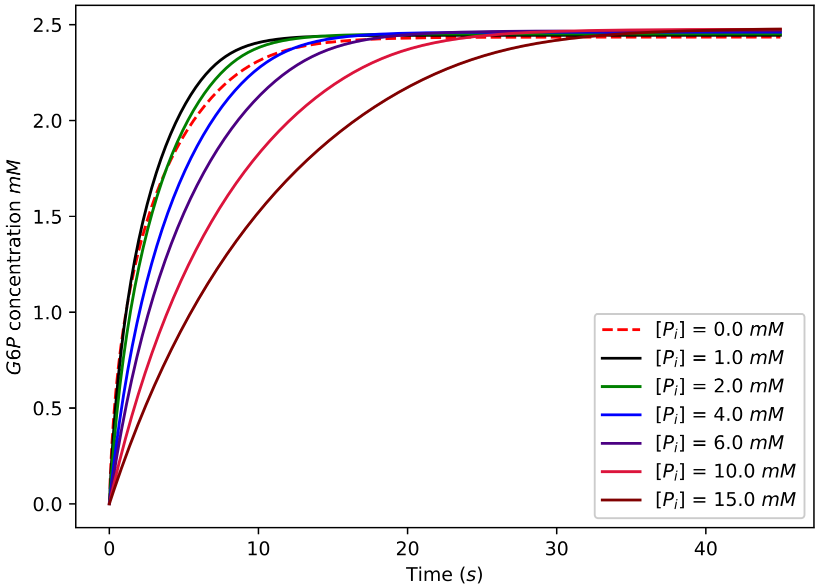
Section 4.2 introduced the mathematical model. Section 4.2.7 described the computational methods used to integrate the model equations, and included some discussion of the numerical method, the choice of parameter values, and the initial conditions. In the current section, we describe some of the numerical results obtained.

The principal purpose of the numerical solutions displayed here is to gain insight into the cellular phosphorylation of glucose by Hexokinase I. To focus attention on the phosphorylation process itself, we make no attempt to model the evolution of intracellular glucose levels. Instead, we simply assume a constant initial concentration of glucose and then track its subsequent conversion via phosphorylation to $G6P$. For similar reasons, we also make no attempt to model the cellular behaviour of $G6P$ subsequent to its production.

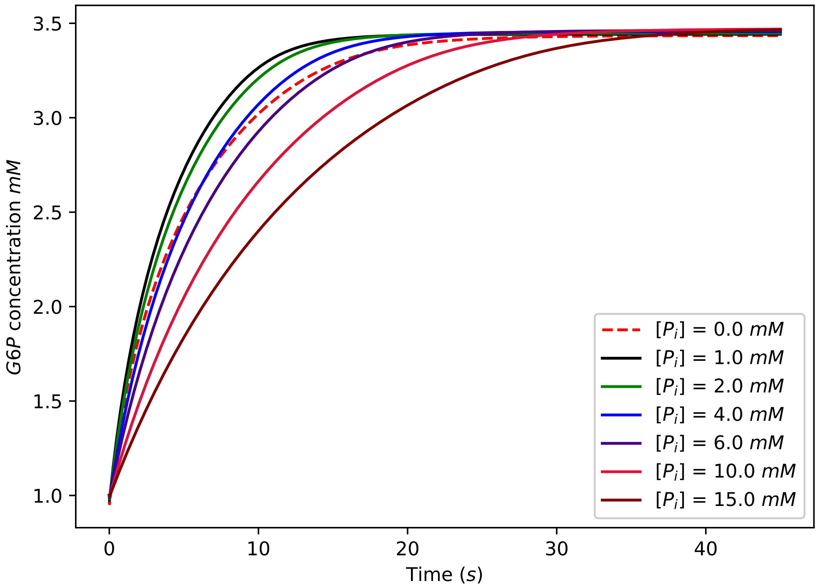
Figure 4.7 shows the time evolution of the concentration of $G6P$ for a range of different initial concentrations of P_i and $G6P$. In the numerical results displayed here, the initial concentration of glucose is taken to be 2.5 mM [188] and the total concentration of the enzyme is taken to be 6.65×10^{-2} mM [187]. We begin by noting some broad features of the behaviour exhibited in Figure 4.7. All of the curves are increasing functions of time, as would be expected since $G6P$ levels increase as the available glucose is phosphorylated. The levelling off of the curves corresponds to the exhaustion of the available glucose substrate. It is also noteworthy that the time scale over which the phosphorylation process is completed is of the order of ten seconds, a prediction that is consistent with literature values [190, 191].

In Figure 4.7, the subplots (a), (b), (c), (d) correspond to differing initial concentrations of $G6P$, with (a) having the lowest initial concentration and (d) the highest. It is clear that the time for the phosphorylation process to be completed increases with increasing initial concentration of $G6P$. This again is as expected since $G6P$ is the species that is responsible for both the allosteric and competitive product inhibition of the enzyme, and so increasing its concentration should slow the phosphorylation process; see Section 4.2.1.

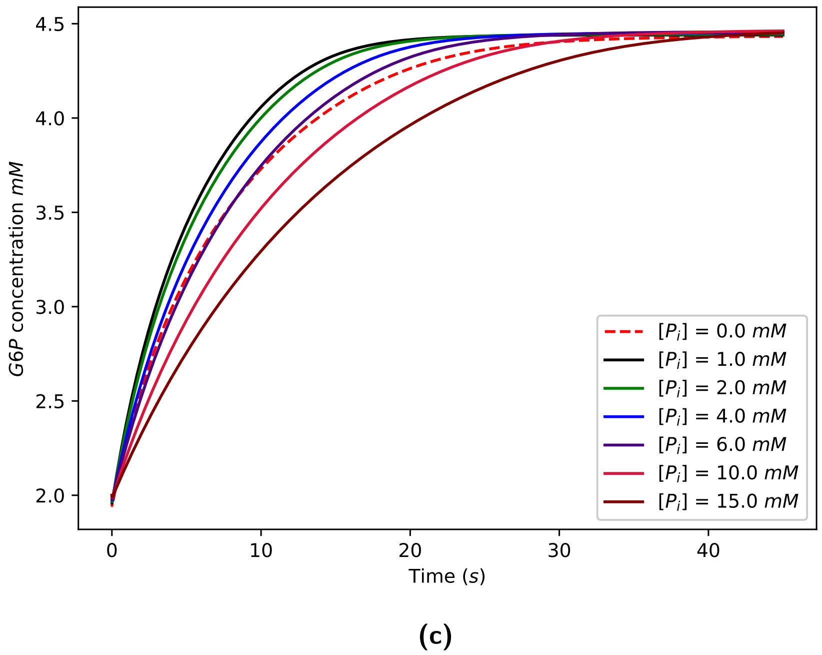
The dependence of the phosphorylation behaviour on the initial concentration of inorganic phosphate is more subtle and interesting. Focus, for example, on the subplot Figure 4.7 (d), and begin by considering the curve corresponding to $[P_i] = 0$. This is the curve corresponding to zero initial P_i concentration, and it gives a convenient reference. We note that for relatively low P_i concentrations (1 mM, 2 mM), the phosphorylation process is faster than the phosphate free case. However, for the higher concentrations ($P_i \geq 10$ mM), we note that the phosphorylation rate is slower relative to the phosphate free case. This is in line with experimental findings [149, 156, 182] which show that for low concentrations of phosphate (few milimolar) enzyme inhibition


2.5
2.0
1.5
1.0
0.5
0.0
G6P concentration mM
0
10
20
30
40
Time (s)
$[P_i]$ = 0.0 mM
$[P_i]$ = 1.0 mM
$[P_i]$ = 2.0 mM
$[P_i]$ = 4.0 mM
$[P_i]$ = 6.0 mM
$[P_i]$ = 10.0 mM
$[P_i]$ = 15.0 mM

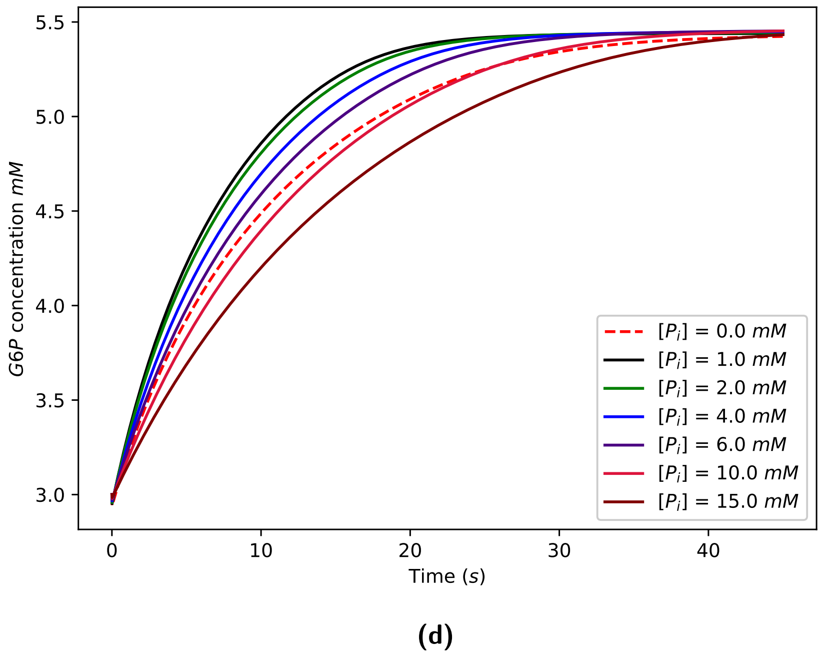
(a)


3.5
3.0
2.5
2.0
1.5
1.0
G6P concentration mM
0
10
20
30
40
Time (s)
$[P_i]$ = 0.0 mM
$[P_i]$ = 1.0 mM
$[P_i]$ = 2.0 mM
$[P_i]$ = 4.0 mM
$[P_i]$ = 6.0 mM
$[P_i]$ = 10.0 mM
$[P_i]$ = 15.0 mM

(b)



(c)



(d)

Figure 4.7: Numerical solutions of the model equations described in Section 4.2. The graphs show the concentration of $G6P$ as a function of time for various initial concentrations of P_i and $G6P$. The initial concentrations for P_i are given in the legends on the graphs, and the initial $G6P$ concentrations are given by (a) 0 mM, (b) 1.0 mM, (c) 2.0 mM, and (d) 3.0 mM. The remaining parameter values can be found in Section 4.2.7.

by $G6P$ is antagonised, and that for higher phosphate concentrations, enzyme activity is inhibited. In the context of the current modelling, this behaviour is explained by recalling that for low concentrations of P_i, the higher affinity N binding site for P_i dominates and inhibition by $G6P$ is antagonised. However, for higher P_i concentrations, P_i binding to the lower affinity C site is significant and enzyme activity is inhibited. This phenomenon is clearly exhibited in Figure 4.8, where we show plots of the phosphorylation rate as a function of inorganic phosphate P_i concentration, and for four different initial concentrations of $G6P$.

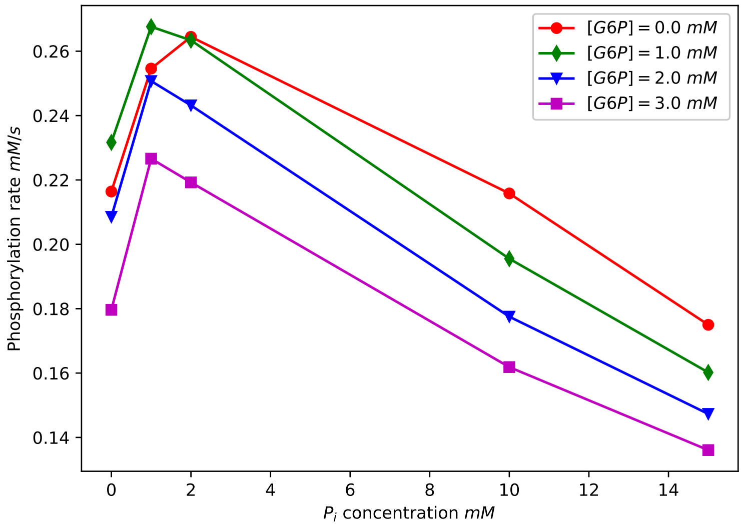
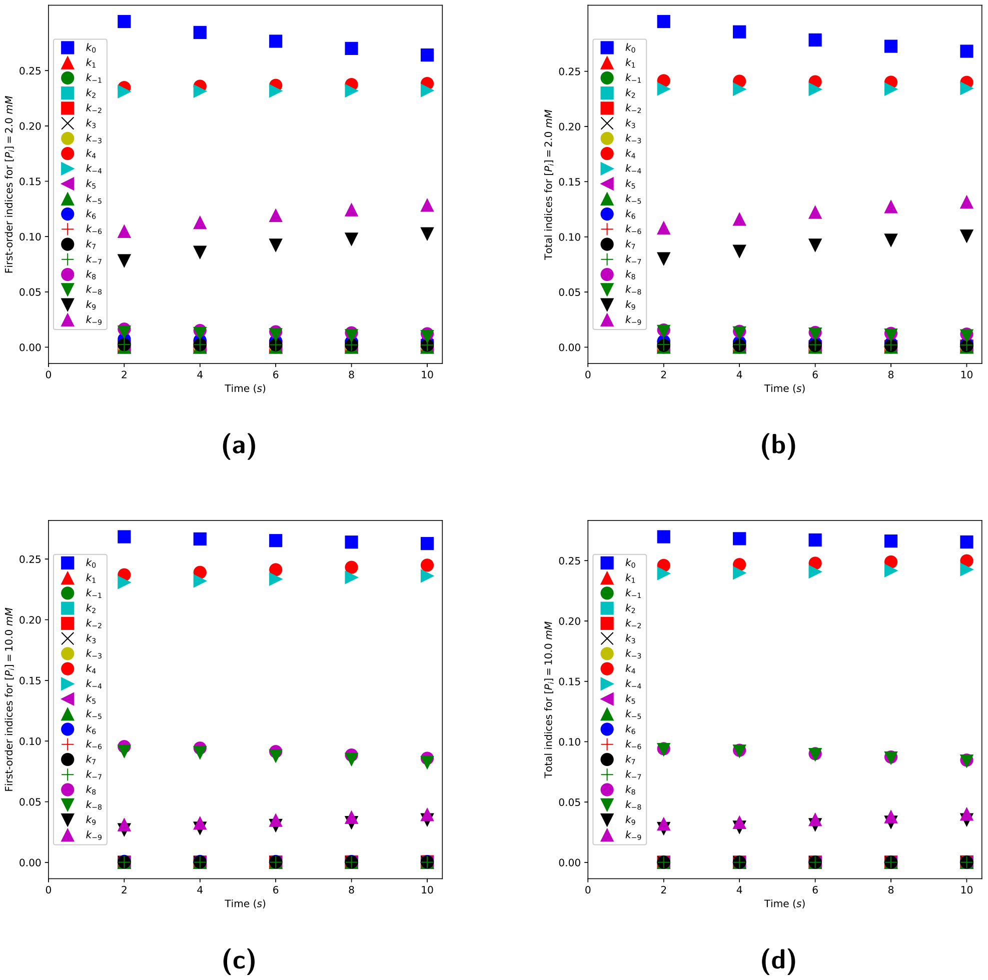


Figure 4.8: Plots of the phosphorylation rate as a function of the initial concentration of phosphate P_i and for four different initial concentrations of $G6P$. The parameter values used to generate these curves can be found in Table 4.2 and Table 4.3. We note an initial increase in the phosphorylation rate in all cases, followed by a subsequent decrease in the rate. This is discussed further in the main text.

4.3.2 Results of the global sensitivity analysis

The Sobol global sensitivity analysis employed in the current study is described in Section 4.2.7, and it was implemented using the SALib package [83]. A Sobol analysis enables us to quantify how variations in the model parameters $\mathbf{p} = (k_i), -9 \leq i \leq 9$ affect the model output. The k_i here are the model rate constants and are described in Table 4.2. The model is of the form $\mathbf{y} = \mathbf{f}(\mathbf{p}, t)$, where the inputs are the parameters $\mathbf{p}$ and the time t, and the output $\mathbf{y}$ give the predictions for the concentrations of the model species at time t. In the current study, we confine our attention to the output for the $G6P$ concentration since $G6P$ is the product here.

Default settings for the SALib package were used, with one exception - no second-order indices were calculated [83]. We calculated first-order and total sensitivity indices for the times $t = 2, 4, 6, 8, 10$ s. The output is then $G6P(t = 2, 4, 6, 8, 10\text{ s})$ and the purpose of the analysis is to evaluate the sensitivity of this output to variations in the parameters k_i using the sensitivity indices. The $G6P$ values were calculated by numerically integrating the governing ordinary differential equations, as previously described. The initial concentration for $G6P$ was taken to be 2.0 mM when numerically integrating the differential equations. Two choices for the initial concentration of phosphate were made, 2.0 mM (low P_i concentration) and 10.0 mM (high P_i concentration). The initial concentrations for the enzyme, glucose and ATP are given in Table 3. The remaining species had zero initial concentration.



(a) **(b)**

(c) **(d)**

Figure 4.9: (a) First-order sensitivity indices ($S1$) and (b) total sensitivity indices (ST) with $[P_i](t = 0) = 2.0\ mM$. (c) First-order sensitivity indices and (d) total sensitivity indices with $[P_i](t = 0) = 10.0\ mM$. The indices are calculated at the times $t = 2, 4, 6, 8, 10$ s. The remaining parameter values are given in the main body of the text.

Figure 4.9 displays first-order sensitivity indices and total sensitivity indices for the model parameters. Figure 4.9 (a) and (b) show values for a low initial phosphate concentration (2.0 mM), while Figure 4.9 (c) and (d) give values for a high initial phosphate concentration (10.0 mM). For convenience, we split the model parameters into three groups - Group I: k_1, k_{-1}, k_2, k_{-2}, k_3, k_{-3}, k_5, k_{-5}, k_6, k_{-6}, k_7, k_{-7}, Group II: k_8, k_{-8}, and Group III: k_0, k_4, k_{-4}, k_9, k_{-9}. The sensitivity indices for all of the Group I parameters are small. This means that the rate of $G6P$ production is relatively insensitive to modest variations in the assumed values of these parameters. The parameters in Group I describe, among other things, the rate of binding and unbinding of glucose to both the N and the C domains of the enzyme, and the rate of binding/unbinding of P_i to the N domain of the enzyme.

We now turn our attention to the parameters in Group III. The first-order sensitivity indices for all of these parameters are relatively large, implying that the $G6P$ production rate is relatively sensitive to variations in the assumed values of these parameters. The Group III parameters determine the turnover rate of the enzyme, the rate of binding/unbinding of ATP to the C domain of the enzyme, and the rate of binding/unbinding of $G6P$ to the C domain of the enzyme when a P_i molecule is bound at its N site. It is noteworthy that the indices S_9 and S_{-9} are quite sensitive to the phosphate concentration, being significantly larger for lower initial phosphate concentration. Hence, for low phosphate concentrations, the binding of P_i to the N binding site is one of the key regulators of enzyme activity.

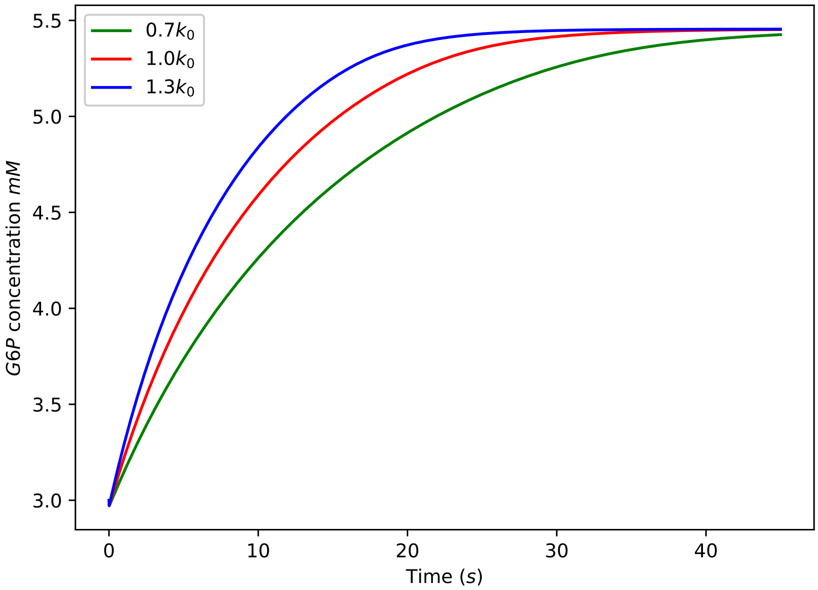
The parameters for Group II (k_{-8}, k_8) are also seen to be sensitive to the phosphate concentration, being small for the low phosphate cases, and significant for higher phosphate cases. These parameters determine the rate of binding/unbinding of P_i to its C binding site.

It is clear from Figure 4.9, that the first-order sensitivity indices are close to the total sensitivity indices. Hence interactions between any of the parameters k_i and the other model parameters do not significantly affect the model output.

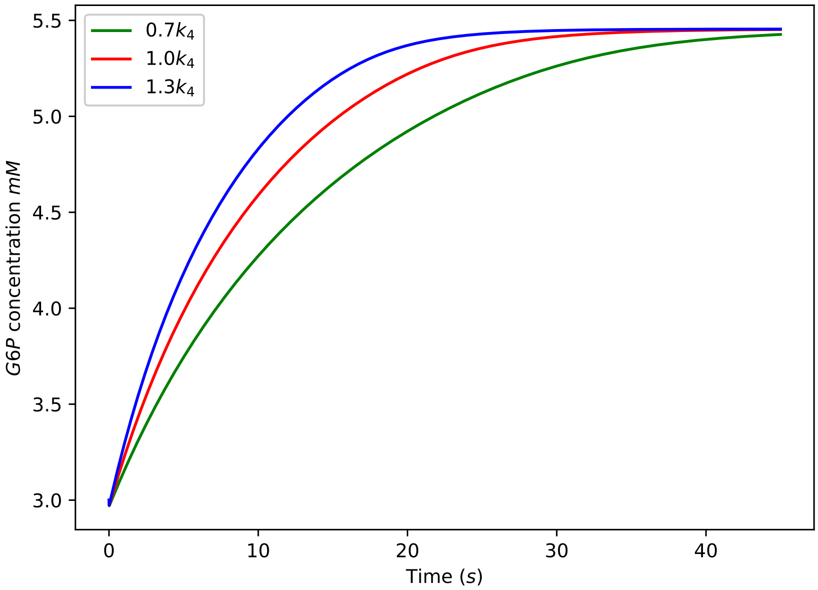
4.3.3 Further numerical results

We now display some further numerical solutions inspired by the results of the sensitivity analysis just presented. In these calculations, the default values used for the parameters k_i are given in Table 4.2, and the initial concentrations for the enzyme, glucose and ATP are given in Table 4.3. The initial concentrations of $G6P$ and P_i were taken to be 3 mM and 6 mM, respectively, corresponding to a high phosphate concentration case.

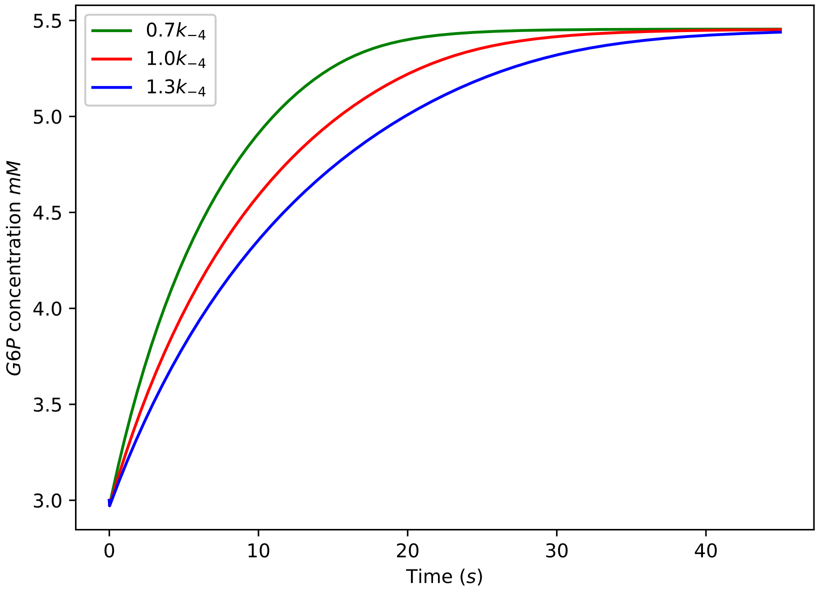
We illustrate how the new numerical results shown in Figure 4.10 were generated by considering a particular example. Consider the curves displayed in Figure 4.10 (a). The middle curve was generated using the default values for the parameters k_i. The upper curve was generated using the default values, except that $1.3k_0$ was used rather than k_0, and the lower curve was generated using $0.7k_0$ rather than k_0. Hence the three curves shown in Figure 4.10 (a) help evaluate the sensitivity of the model output to variations in the parameter k_0.


0.7k_0
1.0k_0
1.3k_0
G6P concentration mM
5.5
5.0
4.5
4.0
3.5
3.0
0
10
20
30
40
Time (s)

(a)

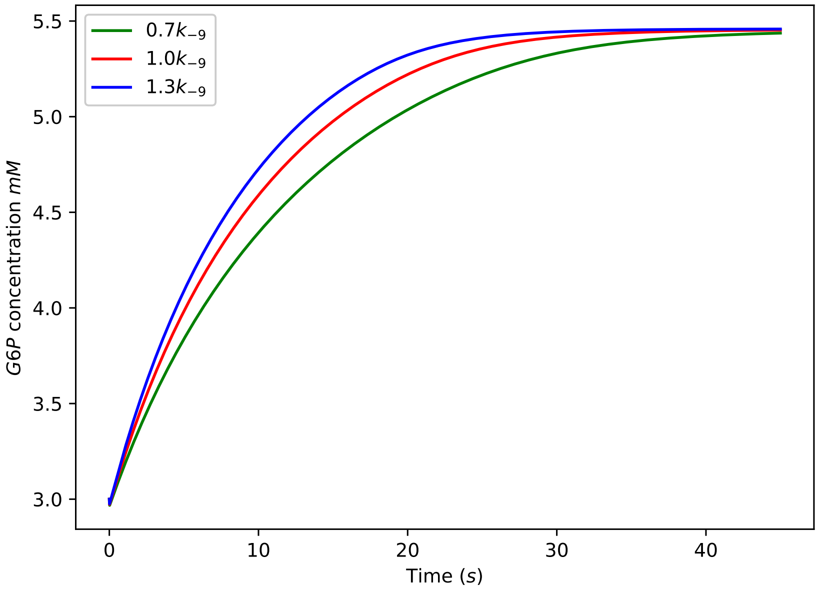

0.7k_4
1.0k_4
1.3k_4
G6P concentration mM
5.5
5.0
4.5
4.0
3.5
3.0
0
10
20
30
40
Time (s)

(b)



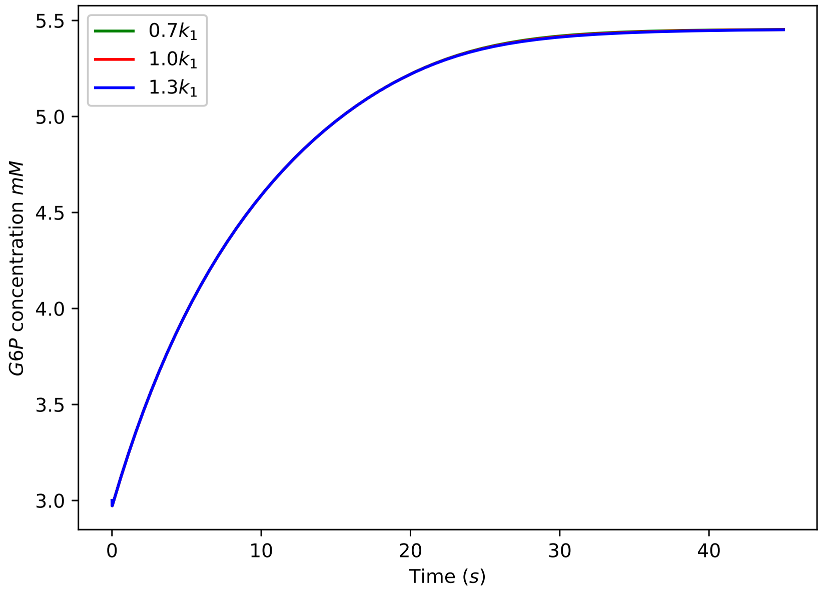
$0.7k_{-4}$
$1.0k_{-4}$
$1.3k_{-4}$
G6P concentration mM
5.5
5.0
4.5
4.0
3.5
3.0
0
10
20
30
40
Time (s)

(c)

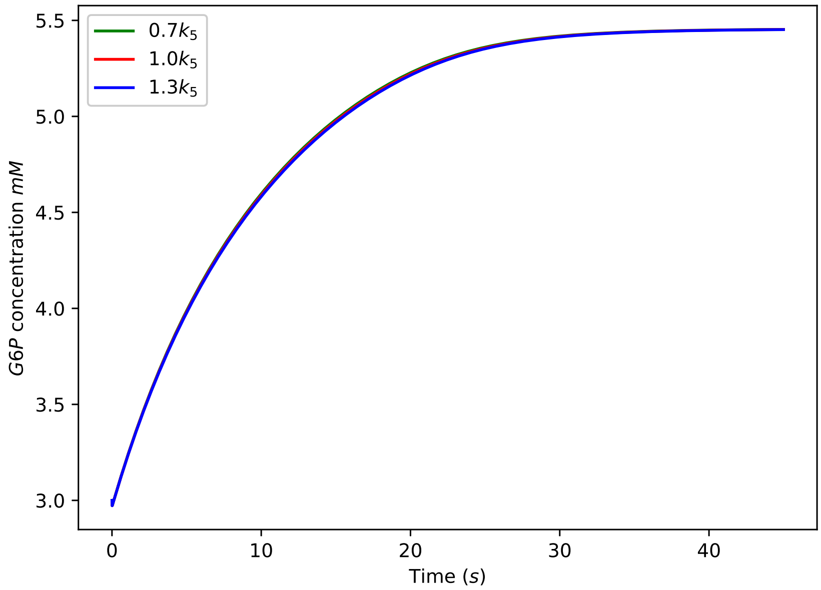


$0.7k_{-9}$
$1.0k_{-9}$
$1.3k_{-9}$
G6P concentration mM
5.5
5.0
4.5
4.0
3.5
3.0
0
10
20
30
40
Time (s)

(d)



(e)



(f)

Figure 4.10: Numerical solutions of the model equations described in Section 4.2. The graphs show the concentration of $G6P$ as a function of time, and the parameter values used can be found in the main body of the text. The solutions displayed help evaluate the sensitivity of the model output to the parameter (a) k_0, (b) k_4, (c) k_{-4}, (d) k_{-9}, (e) k_1, and (f) k_5.

The remaining subplots in Figure 4.10 are generated by repeating this process for the parameters k_4, k_{-4}, k_{-9}, k_1, and k_5.

The results shown in Figure 4.10 are consistent with the predictions of the sensitivity analysis of Section 4.3.2 since the model output is seen to be quite sensitive to the parameters with relatively large sensitivity indices (k_0, k_4, k_{-4}, k_{-9}), but insensitive to the parameters with small indices (k_1, k_5).

4.3.4 Model reduction

Motivated by the results of the sensitivity analysis in Section 4.3.2, we now consider a simplified model (SM). This model is obtained from the full model described in Section 4.2 by setting

$$k_1 = k_{-1} = k_3 = k_{-3} = 0.$$

In this SM model, glucose molecules do not bind to their N domain site, and ATP molecules do not bind to their N domain site.

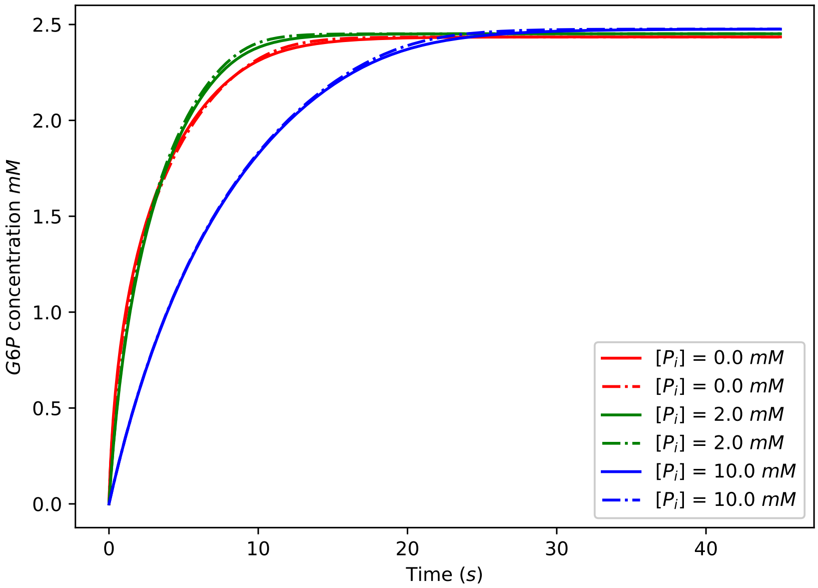
We have previously seen that the sensitivity indices for all of these parameters are small, and so we anticipate that the output for these models should typically closely match that for the full model. The initial conditions used to generate the numerical solutions for the SM model are the same as those used for the full model, with the exception of the initial conditions for $G6P$ and P_i, which are specified on the numerical figures. Figure 4.11 compares solutions of the full model with solutions of the SM, where solid curves are solutions to the full model, and dashed-dotted curves are solutions to the SM.

It is seen in Figure 4.11 that the results of the full model and the SM are close in all cases, suggesting that the full model may reasonably be simplified by dropping the mechanisms of glucose and ATP binding to the N domains.

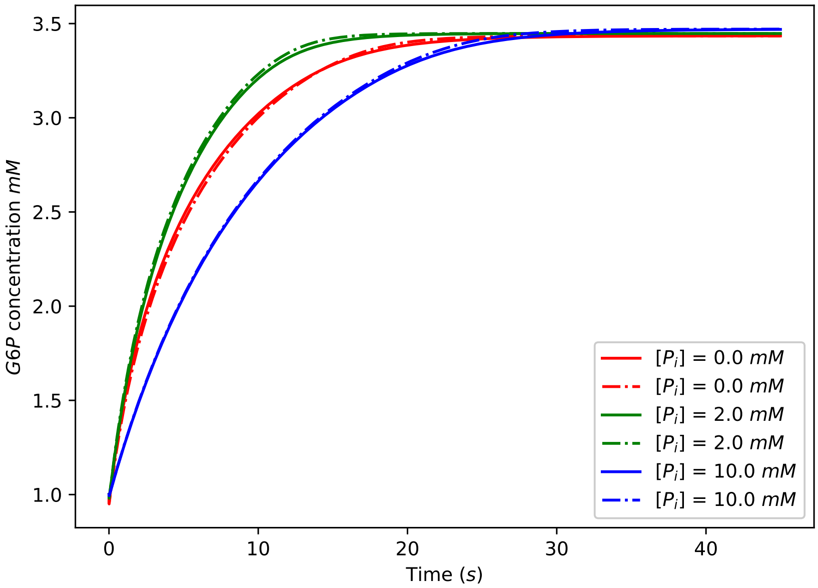
Intuitively, we can explain why the original model output is rather insensitive to the parameters for the N binding sites for glucose (k_1, k_{-1}) and ATP (k_3, k_{-3}) as follows.

- The reason why glucose binding to the N domain does not significantly affect the phosphorylation rate is because the concentration of glucose is typically much larger than the concentration of enzyme. Hence, glucose binding to the N domain does not significantly reduce the free concentration of glucose, and there is ample glucose remaining to participate in phosphorylation.

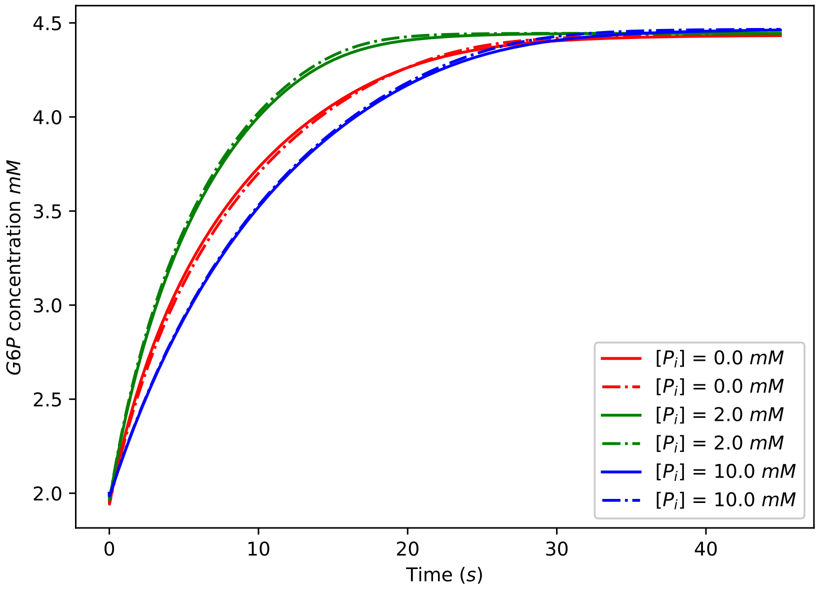
- The reason ATP binding to the N domain does not significantly affect the phosphorylation rate is more complex since ATP binding to this domain has a number of effects. Firstly, as for glucose, ATP concentration is typically much higher than that of the enzyme, so that ATP binding at the N domain does not significantly reduce the ATP pool available


G6P concentration mM
0.0
0.5
1.0
1.5
2.0
2.5
0
10
20
30
40
Time (s)
[P_i] = 0.0 mM
[P_i] = 0.0 mM
[P_i] = 2.0 mM
[P_i] = 2.0 mM
[P_i] = 10.0 mM
[P_i] = 10.0 mM

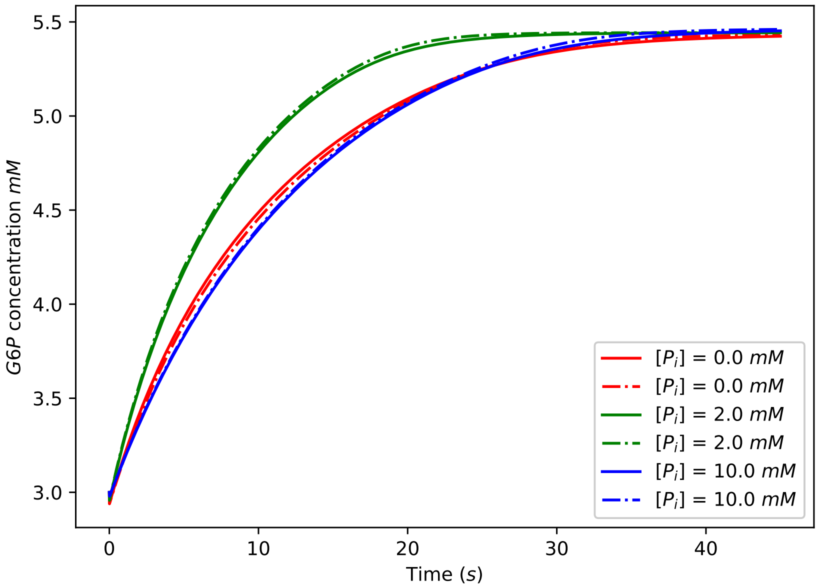
(a)


G6P concentration mM
1.0
1.5
2.0
2.5
3.0
3.5
0
10
20
30
40
Time (s)
[P_i] = 0.0 mM
[P_i] = 0.0 mM
[P_i] = 2.0 mM
[P_i] = 2.0 mM
[P_i] = 10.0 mM
[P_i] = 10.0 mM

(b)



(c)



(d)

Figure 4.11: Comparison of numerical solutions to the full model and the Simplified Model (SM). The solid curves are solutions to the full model and the dashed-dotted curves are solutions to the SM. The initial concentrations for the phosphate P_i are given on the figures, and the initial concentrations for $G6P$ are given by (a) 0 mM, (b) 1.0 mM, (c) 2.0 mM, and (d) 3.0 mM. The remaining parameter values used can be found in the main body of the text.

for phosphorylation. Also, ATP competes with both $G6P$ and P_i for the N domain binding sites. Competition with $G6P$ reduces enzyme inhibition (as previously explained), whereas competition with P_i increases inhibition (again, as previously explained), and we speculate that there is a cancellation effect here.

4.4 Conclusions

In this chapter, we have developed a comprehensive mathematical model describing the phosphorylation of glucose by the enzyme Hexokinase I. Glucose phosphorylation is the first step of the glycolysis pathway, and so it is carefully regulated by cells. The regulation of hexokinase I is quite complex and includes three inhibitory mechanisms: a competitive product inhibitory mechanism, an allosteric inhibitory mechanism, and a competitive inhibitory mechanism. We used the mathematical model to help unpick the regulatory behaviour of Hexokinase I. In particular, we obtained the following results.

- *Numerical simulations.* The model was numerically integrated using the SciPy Python library, and the solutions obtained were found to be consistent with the known behaviour of hexokinase I. For example, it was found that the rate of phosphorylation decreased with increasing concentration of $G6P$. Also, it was found that low phosphate concentrations antagonise hexokinase I inhibition, while high phosphate concentrations inhibit hexokinase I.

- *Global sensitivity analysis.* A global sensitivity analysis of the model was implemented to help identify the key mechanisms of hexokinase I regulation. The results of this analysis indicate that the rate of phosphorylation is quite sensitive to the following factors: the turnover rate of the enzyme; the rate of binding/unbinding of ATP to/from the C domain of the enzyme; the rate of binding/unbinding of $G6P$ to/from the C domain of the enzyme with a P_i molecule bound at the N domain for low phosphate concentration; and the rate of binding/unbinding of phosphate to/from the C domain of the enzyme for high phosphate concentration.

- *Simplified model.* One reduced model was developed based on the results of the sensitivity analysis. This simpler model produces results that closely match the results of the full model.

- *Software.* The software developed in this chapter to numerically integrate the governing equations and to implement the sensitivity analysis has been made available in the Appendix B and online.

Although the model developed in the current chapter is comprehensive and detailed, there is some scope for improvement. For example, the full detail of the Bi Bi mechanism could be incorporated in the modelling. Also, glucose-6-phosphate binding to its N binding site not only allosterically inhibits the enzyme but also stimulates enzyme release from mitochondria [192, 193], and we have made no attempt to describe this release behaviour. Finally, the possible inhibition of hexokinase I by ADP [185] has not been explored in the current study.

Chapter 5

Mathematical models for enzymatic inhibition by product

In the previous chapter, we developed a detailed mathematical model describing the phosphorylation of glucose by Hexokinase I. The model was complex, and incorporated three mechanisms for enzyme inhibition. However, the complexity of this model made it impossible to obtain simple analytical expressions for the rate of product formation. It also made it difficult to obtain simple qualitative insights into the enzyme behaviour. In an effort to overcome those deficiencies, we shall consider two simpler related models in this chapter. The first model focuses on the mechanism of competitive product inhibition only, while the second model considers allosteric inhibition only. For each of these models, we develop expressions for the rate of product formation. We also explain how the models can, in appropriate circumstances, model the phosphorylation of glucose by hexokinase I.

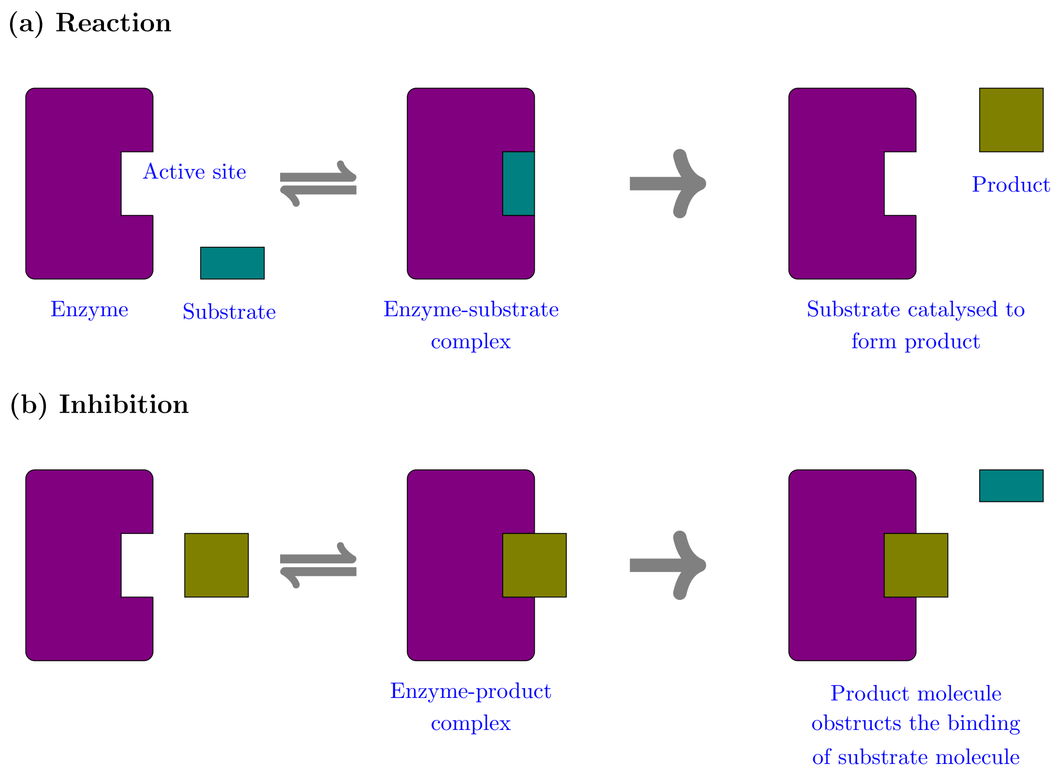
5.1 Model I: Competitive product inhibition

In this section, we develop a mathematical model for the action of an enzyme that is inhibited by its product. Although the formula for the product formation rate is already available in the literature [187, 194, 195], we shall derive it in a transparent manner by non-dimensionalising the equations and making rational approximations.

A minimal set of chemical reactions representing competitive product inhibition is given by

$$\begin{aligned} S + E \;&\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}\; ES \xrightarrow{k_0} E + P, \\ P + E \;&\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}\; EP, \end{aligned}$$

where S, E, P, ES, and EP represent substrate, free enzyme, product, enzyme-substrate complex, and enzyme-product complex, respectively; see Figure 5.1.


(a) Reaction

Active site
Product
Enzyme
Substrate
Enzyme-substrate complex
Substrate catalysed to form product

(b) Inhibition

Enzyme-product complex
Product molecule obstructs the binding of substrate molecule

Figure 5.1: *Model I. Product inhibition.* Diagram of reactions and inhibition. The enzyme here has one binding site that can accommodate both the substrate and the product. In (a) a substrate binds to a free enzyme molecule to form an enzyme-substrate complex. The enzyme then catalyses the substrate to form a product. In (b) binding of a product molecule to a free enzyme molecule to form an enzyme-product complex prevents the enzyme molecule from binding with a substrate.

Applying the law of mass action in the usual way, the corresponding governing ordinary differential equations are given by

$$\begin{aligned}
\frac{d[E]}{dt} &= (k_0 + k_{-1})[ES] + k_{-2}[EP] - k_1[E][S] - k_2[E][P],\\
\frac{d[ES]}{dt} &= k_1[E][S] - (k_0 + k_{-1})[ES],\\
\frac{d[EP]}{dt} &= k_2[E][P] - k_{-2}[EP],\\
\frac{d[S]}{dt} &= k_{-1}[ES] + k_1[E][S],\\
\frac{d[P]}{dt} &= k_0[ES] + k_{-2}[EP] - k_2[E][P],
\end{aligned} \tag{5.1}$$

where $[X] = [X](t)$ denotes the concentration of species X at time t. These equations are to be solved subject to the initial conditions

$$[E](t=0) = e_0, \;\; [S](t=0) = s_0,$$

$$[ES](t=0) = 0, \;\; [EP](t=0) = 0, \;\; [P](t=0) = 0,$$

where e_0, s_0 are positive constants corresponding to the initial concentrations of enzyme and substrate, respectively. Forming $(5.1)_1$+$(5.1)_2$+ $(5.1)_3$ and integrating yields

$$[E] + [ES] + [EP] = e_0, \tag{5.2}$$

which is an expression of conservation of enzyme.

Introducing the dimensionless variables

$$e = \frac{[E]}{e_0},\ c_1 = \frac{[ES]}{e_0},\ c_2 = \frac{[EP]}{e_0},\ s = \frac{[S]}{s_0},\ p = \frac{[P]}{s_0},\ \tau = e_0 k_1 t,$$

the governing equations may be written in the equivalent dimensionless form

$$\begin{aligned}
&e + c_1 + c_2 = 1,\\
&\varepsilon\frac{dc_1}{d\tau} = -(s + \hat{k}_0 + \hat{k}_{-1})c_1 - sc_2 + s,\\
&\varepsilon\frac{dc_2}{d\tau} = \hat{k}_2\left(-pc_1 - (p + \hat{k}_{-2}/\hat{k}_2)c_2 + p\right),\\
&\frac{ds}{d\tau} = \hat{k}_{-1}c_1 - s(1 - c_1 - c_2)),\\
&\frac{dp}{d\tau} = \hat{k}_0 c_1 + \hat{k}_{-2}c_2 - \hat{k}_2 p(1 - c_1 - c_2),
\end{aligned} \tag{5.3}$$

where

$$\varepsilon = \frac{e_0}{s_0},\ \hat{k}_0 = \frac{k_0}{k_1 s_0},\ \hat{k}_{-1} = \frac{k_{-1}}{k_1 s_0},\ \hat{k}_2 = \frac{k_2}{k_1},\ \hat{k}_{-2} = \frac{k_{-2}}{k_1 s_0}, \tag{5.4}$$

are dimensionless parameters. These equations are solved subject to the initial conditions

$$e(t = 0) = 1,\ s(t = 0) = 1,\ c_1(t = 0) = 0,\ c_2(t = 0) = 0,\ p(t = 0) = 0. \tag{5.5}$$

In applications, the amount of substrate initially present typically greatly exceeds the enzyme present, so that $e_0 \ll s_0$, or $\varepsilon \ll 1$. Hence it is of value to consider the behaviour of (5.3),(5.5) in the limit $\varepsilon \to 0$. There is an initial transient at $\tau = O(\varepsilon)$ as $\varepsilon \to 0$, but this behaviour is of limited practical interest, and its discussion is omitted here. For $\tau = O(1)$, we have at leading order as $\varepsilon \to 0$ that (see (5.3))

$$e + c_1 + c_2 = 1,\ -(s + \hat{k}_0 + \hat{k}_{-1})c_1 - sc_2 + s = 0,\ -pc_1 - (p + \hat{k}_{-2}/\hat{k}_2)c_2 + p = 0,$$

and these expressions may be manipulated to give

$$c_1 = \frac{s}{s + (\hat{k}_0 + \hat{k}_{-1})(1 + p\hat{k}_2/\hat{k}_{-2})},\ c_2 = \frac{p}{p + (1 + s/(\hat{k}_0 + \hat{k}_{-1}))\hat{k}_{-2}/\hat{k}_2}.$$

Substituting these expressions into (5.3) gives

$$\frac{dp}{d\tau} = \frac{\hat{k}_0 s}{s + (\hat{k}_0 + \hat{k}_{-1})(1 + p\hat{k}_2/\hat{k}_{-2})}. \tag{5.6}$$

Reverting to dimensional variables, the rate of formation of product is now given by

$$v = \frac{d[P]}{dt} = k_1 e_0 s_0 \frac{dp}{d\tau} \tag{5.7}$$

and using (5.6), this leads to

$$v = \frac{d[P]}{dt} = \frac{V_{max}[S]}{[S] + K_m\left(1 + \frac{[P]}{K_D}\right)}, \tag{5.8}$$

where

$$V_{max} = k_0 e_0, \quad K_m = \frac{k_0 + k_{-1}}{k_1}, \quad K_D = \frac{k_{-2}}{k_2}.$$

Here $V_{max} = k_0 e_0$ is the maximum production rate for the enzyme, K_m is the Michaelis constant for the enzyme in the absence of product inhibition, and K_D is the dissociation constant for product binding to the enzyme. Notice that we can write (5.8) as

$$v = \frac{V_{max}[S]}{[S] + K_m^{app}}, \tag{5.9}$$

where

$$K_m^{app} = K_m\left(1 + \frac{[P]}{K_D}\right) \tag{5.10}$$

is the apparent Michaelis-Menten constant that takes account of competitive product inhibition. It is noteworthy here that the maximal production rate for the enzyme V_{max} is unaffected by product inhibition. However, the apparent Michaelis-Menten constant increases linearly with product concentration; see Figure 5.2. Figure 5.3 shows the Lineweaver-Burk plots of the product formation rate formula (5.9) for $[P] = 0$ and one value with $[P] > 0$. It can be seen that the maximal product formation rate does not depend on the product concentration, while the slope of the line corresponding to $[P] > 0$ is greater than that of the line corresponding to $[P] = 0$, as would be expected since the presence of product slows the speed of product formation.

5.1.1 Model I and glucose phosphorylation by mini hexokinase I

The model just described may be applicable to a mini hexokinase I system. A mini hexokinase molecule consists of the C terminal half only of the hexokinase I enzyme [165, 166], and this corresponds to the enzyme species E in our model above; see Figure 5.4. The active site here is the binding site for ATP, so that ATP corresponds to the substrate S in the model. The product P here is $G6P$ since $G6P$ competes with ATP for the ATP binding site. However, the correspondence between the model and the mini hexokinase I system falls down here since $G6P$ is not a direct product of ATP binding - recall that a glucose molecule must also be bound to its site in the C terminal domain in

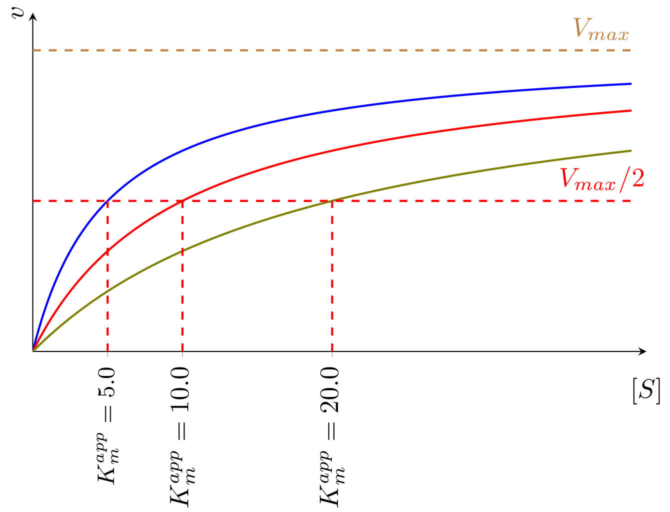


Figure 5.2: *Model I.* Plots of the rate of product formation for the case of competitive product inhibition. The relevant formula is given in equation (5.8), and the plots shown illustrate the effect of the product concentration on the product formation rate. The parameter values used to generate these plots are given by $V_{max} = 4.0\ mM/s$, $K_m = 5.0\ mM$, $K_D = 1.0\ mM$, and $[P] = 0.0, 1.0, 3.0\ mM$, with corresponding values $K_m^{app} = 5.0\ mM$, $K_m^{app} = 10.0\ mM$, $K_m^{app} = 20.0\ mM$, respectively.

order for $G6P$ to be formed. However, $G6P$ would be the effective product of ATP binding if ATP binding is the rate-limiting step for product formation. This would be the case for sufficiently high concentrations of glucose, for example. The model also does not take account of phosphate binding, and so would only apply to the mini hexokinase system if phosphate concentrations are sufficiently low. However, in these circumstances, the rate of production of $G6P$ may be approximated by (see (5.8))

$$v = \frac{V_{max}[ATP]}{[ATP] + K_m^{ATP}(1 + [G6P]/K_D^{G6P})}. \tag{5.11}$$

Figure 5.5 displays plots of this formula for different concentrations of the product $G6P$.

5.2 Model II: Allosteric product inhibition

We now turn attention to a model (Model II) that focuses on the mechanism of allosteric product inhibition of an enzyme. Recall from Chapter 4 that allosteric product inhibition is one of the key mechanisms involved in the phosphorylation of glucose by hexokinase I. The study of Model II will yield insights into this aspect of the phosphorylation of glucose. Allosteric product inhibition is also involved in the regulation of other enzymes, for example, horseradish peroxidase [196], and so it is of practical value to establish a for-

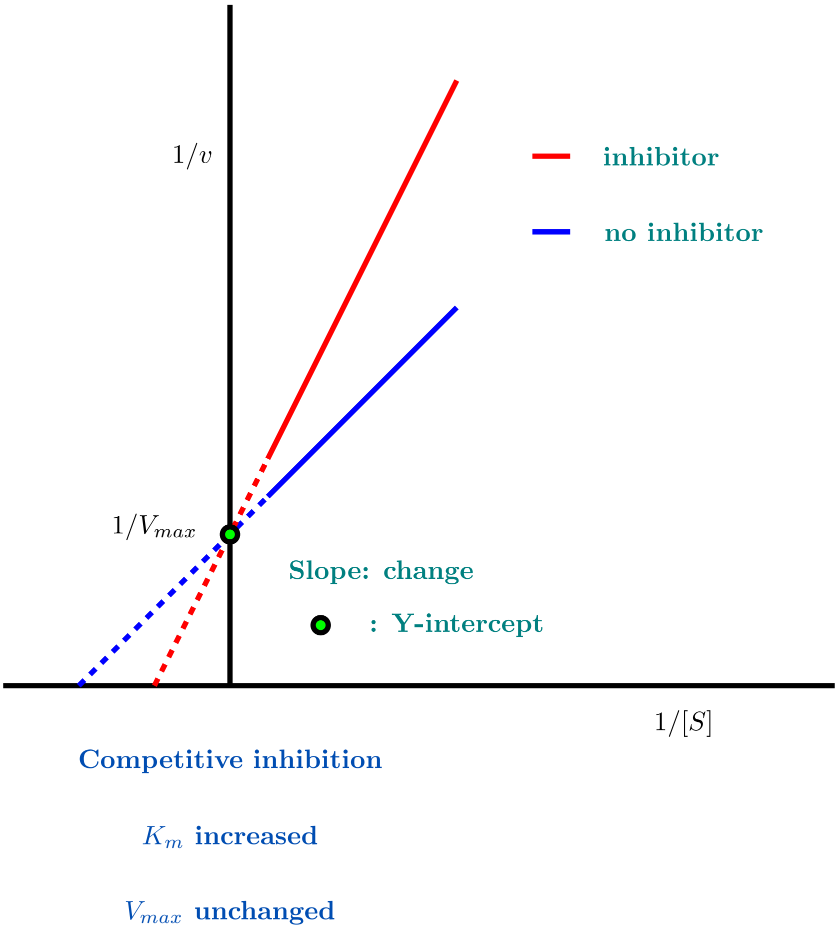


Figure 5.3: *Model I.* Lineweaver-Burk plots of the product formation rate formula (5.9) for $[P] = 0$ and $[P] > 0$.

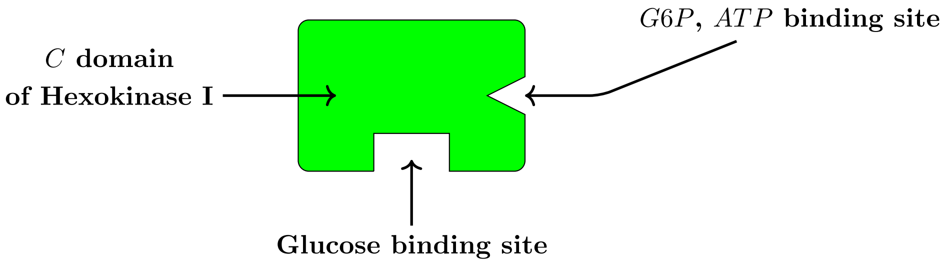


Figure 5.4: Mini hexokinase I.

mula for product formation rate for this mechanism. Hence, in this section, we develop and analyse a minimal model for allosteric inhibition.

Figure 5.6 graphically depicts the model reactions. In the model, the enzyme molecule has two binding sites, one for the substrate (active site) and another for the product (allosteric site). When the product of the active site binds to the allosteric site, it deactivates the active site. The chemical reactions

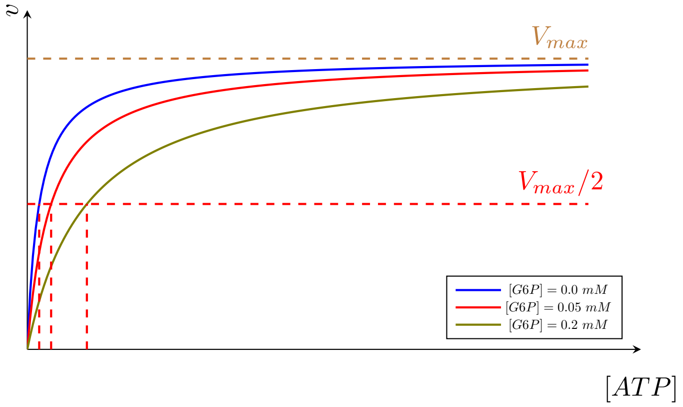


Figure 5.5: Plots of the product formation rate for different concentrations of product $G6P$. The relevant formula is given in equation (5.11). The parameters values used here are $k_0 = 60\ s^{-1}$, $K_m^{ATP} = 0.68\ mM$, $K_D^{G6P} = 0.05\ mM$ [180], $[HK] = 0.05\ mM$, and $[G6P] = 0.0, 0.05, 0.2\ mM$.

for the model are

$$\begin{aligned} S + E &\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_0} E + P, \\ P + E &\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} EP, \\ S + EP &\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ESP, \\ P + ES &\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} ESP, \end{aligned} \tag{5.12}$$

where E, S, and P denote an enzyme molecule, a substrate molecule, and a product molecule, respectively. The complexes ES, EP, and ESP have the obvious interpretation; see Figure 5.6.

The governing ordinary differential equations here are:

$$\begin{aligned} \frac{d[E]}{dt} &= (k_0 + k_{-1})[ES] + k_{-2}[EP] - (k_1[S] + k_2[P])[E], \\ \frac{d[ES]}{dt} &= k_1[E][S] + k_{-2}[ESP] - (k_0 + k_{-1} + k_2[P])[ES], \\ \frac{d[EP]}{dt} &= k_2[E][P] + k_{-1}[ESP] - (k_{-2} + k_1[S])[EP], \\ \frac{d[ESP]}{dt} &= k_1[S][EP] + k_2[P][ES] - (k_{-1} + k_{-2})[ESP], \\ \frac{d[S]}{dt} &= k_{-1}([ES] + [ESP]) - k_1[S]([E] + [EP]), \\ \frac{d[P]}{dt} &= k_0[ES] + k_{-2}([EP] + [ESP]) - k_2[P]([E] + [ES]). \end{aligned} \tag{5.13}$$

and these are solved subject to the initial conditions

$$\begin{aligned} &[E](t = 0) = e_0,\ [P](t = 0) = 0,\ [EP](t = 0) = 0, \\ &[S](t = 0) = s_0,\ [ES](t = 0) = 0,\ [ESP](t = 0) = 0, \end{aligned}$$

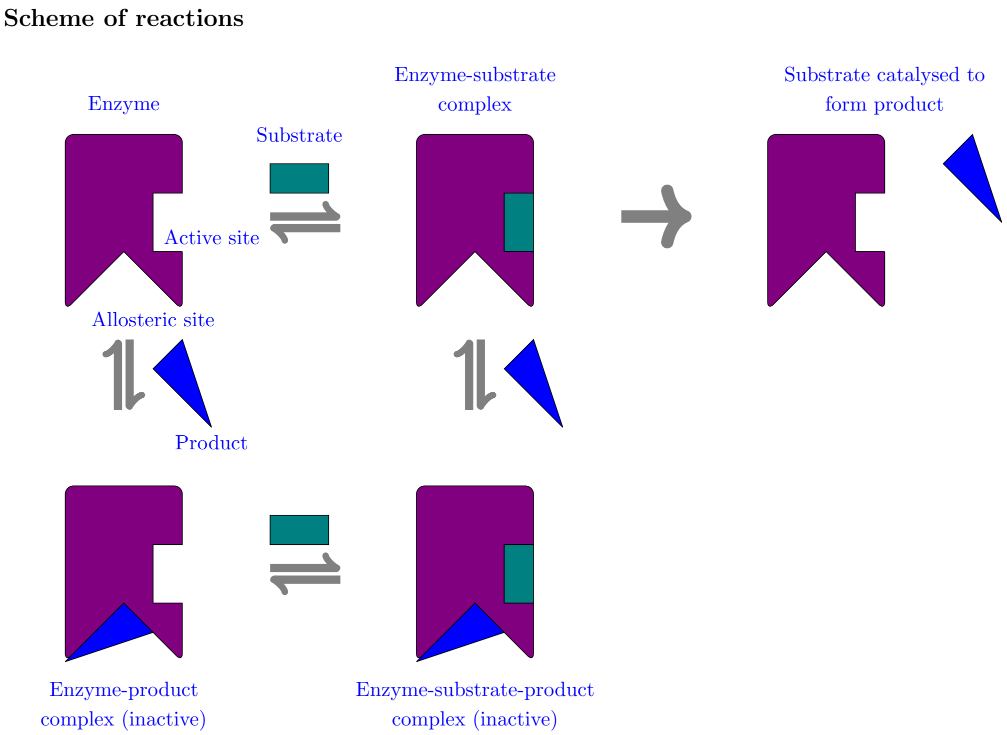

Scheme of reactions
Enzyme
Substrate
Enzyme-substrate complex
Substrate catalysed to form product
Active site
Allosteric site
Product
Enzyme-product complex (inactive)
Enzyme-substrate-product complex (inactive)

Figure 5.6: *Model II. Allosteric inhibition.* Diagram of reactions and inhibitions. The top row corresponds to the reactions $(5.12)_1$, while the bottom row corresponds to the reactions $(5.12)_3$. The left and right vertical reactions corresponds to $(5.12)_2$, and $(5.12)_4$, respectively.

where e_0 and s_0 are the constant initial concentrations of enzyme and substrate, respectively. Adding the first four equations in (5.13) and integrating gives

$$[E] + [ES] + [EP] + [ESP] = e_0, \tag{5.14}$$

which corresponds to conservation of enzyme.

We non-dimensionalize the equations by introducing the dimensionless quantities

$$\begin{aligned} &e = \frac{[E]}{e_0},\ c_1 = \frac{[ES]}{e_0},\ c_2 = \frac{[EP]}{e_0},\ c_3 = \frac{[ESP]}{e_0}, \\ &s = \frac{[S]}{s_0},\ p = \frac{[P]}{s_0},\ \tau = e_0 k_1 t,\ \varepsilon = \frac{e_0}{s_0}, \end{aligned}$$

to obtain the dimensionless equations

$$\begin{aligned}
\varepsilon\frac{dc_1}{d\tau} &= -(\hat{k}_0+\hat{k}_{-1}+s+\hat{k}_2p)c_1 - sc_2 + (\hat{k}_{-2}-s)c_3 + s,\\
\varepsilon\frac{dc_2}{d\tau} &= -\hat{k}_2pc_1 - (\hat{k}_{-2}+s+\hat{k}_2p)c_2 + (\hat{k}_{-1}-\hat{k}_2p)c_3 + \hat{k}_2p,\\
\varepsilon\frac{dc_3}{d\tau} &= \hat{k}_2pc_1 + sc_2 - (\hat{k}_{-1}+\hat{k}_{-2})c_3, \qquad (5.15)\\
\frac{ds}{d\tau} &= (\hat{k}_{-1}-s)(c_1+c_3) - s,\\
\frac{dp}{d\tau} &= \hat{k}_0c_1 + (\hat{k}_{-2}+\hat{k}_2p)(c_2+c_3) - \hat{k}_2p,
\end{aligned}$$

where

$$\hat{k}_0 = \frac{k_0}{k_1s_0}, \quad \hat{k}_{-1} = \frac{k_{-1}}{k_1s_0}, \quad \hat{k}_2 = \frac{k_2}{k_1}, \quad \hat{k}_{-2} = \frac{k_{-2}}{k_1s_0}.$$

We have omitted the equation for e here since this can be determined from the dimensionless form for (5.14), given by

$$e + c_1 + c_2 + c_3 = 1.$$

Under typical conditions, the initial concentration of substrate is much larger than the substrate concentration, so that $e_0 \ll s_0$, or $\varepsilon \ll 1$. Taking the limit $\varepsilon \to 0$ in the equations (5.15), we obtain at leading order that (for $\tau = O(1)$)

$$\begin{aligned}
c_1 &= \frac{\hat{k}_{-2}s(s+\hat{k}_2p+\hat{k}_{-1}+\hat{k}_{-2})}{(\hat{k}_{-2}+\hat{k}_2p)(s^2+as+b)}, \quad c_2 = \frac{\hat{k}_2p(\hat{k}_{-1}s+b)}{(\hat{k}_{-2}+\hat{k}_2p)(s^2+as+b)},\\
c_3 &= \frac{\hat{k}_2ps(s+a-\hat{k}_{-1})}{(\hat{k}_{-2}+\hat{k}_2p)(s^2+as+b)},
\end{aligned}$$

where

$$a = \hat{k}_2p + \hat{k}_0 + 2\hat{k}_{-1} + \hat{k}_{-2}, \quad b = \hat{k}_{-1}(\hat{k}_2p + \hat{k}_0 + \hat{k}_{-1} + \hat{k}_{-2}) + \hat{k}_0\hat{k}_{-2}.$$

Substituting these expressions into $(5.15)_5$ gives

$$\frac{dp}{d\tau} = \frac{\hat{k}_{-2}\hat{k}_0s(s+\hat{k}_2p+\hat{k}_{-1}+\hat{k}_{-2})}{(\hat{k}_{-2}+\hat{k}_2p)(s^2+as+b)},$$

and reverting to dimensional variables gives

$$v = \frac{d[P]}{dt} = k_1e_0s_0\frac{dp}{d\tau},$$

which leads to

$$v = \frac{V_{max}}{1+[P]/K_{D,P}}\frac{[S]^2+A[S]}{[S]^2+B[S]+C}, \qquad (5.16)$$

where

$$\begin{aligned}A &= K_{D,S} + (1 + [P]/K_{D,P})k_{-2}/k_1,\\ B &= K_{D,S} + K_m + (1 + [P]/K_{D,P})k_{-2}/k_1,\\ C &= K_{D,S}[K_m + (1 + [P]/K_{D,P})k_{-2}/k_1] + k_0k_{-2}/k_1^2,\end{aligned} \tag{5.17}$$

and

$$V_{max} = k_0e_0, \quad K_{D,S} = \frac{k_{-1}}{k_1}, \quad K_{D,P} = \frac{k_{-2}}{k_2}, \quad K_m = \frac{k_0 + k_{-1}}{k_1}.$$

Letting $[S] \to \infty$ in (5.16) gives

$$v \to \frac{V_{max}}{1 + [P]/K_{D,P}},$$

and so the maximal rate of product formation is decreased by the factor $(1 + [P]/K_{D,P})$. Hence, for allosteric inhibition, the maximal rate of product formation is reduced by the concentration of product. This contrasts with the case of competitive product inhibition discussed earlier, where it was seen that the maximal rate of product formation is independent of the product concentration (equation (5.8)).

We can define an apparent Michaelis-Menten constant for the system, K_m^{app} via the equation

$$v\left([S] = K_m^{app}\right) = \frac{1}{2}\left(\frac{V_{max}}{1 + [P]/K_{D,P}}\right).$$

Using (5.16), this leads to

$$K_m^{app} = \frac{1}{2}\left(K_m - A + \sqrt{(K_m - A)^2 + 4C}\right), \tag{5.18}$$

where A, C are given in (5.17). This dependence is quite complex, but there are various simpler limits of interest that may be considered. For example, if $K_m \gg A, \sqrt{C}$, we have $K_m^{app} \sim K_m$. Also, if $A - K_m \gg \sqrt{C}$, we have $K_m^{app} \sim C/(A - K_m)$ - if we then consider the further limit, $[P] \gg K_{D,P}$, we arrive at $K_m^{app} \sim K_{D,S}$; see Figure 5.7. In this figure, we observe that the apparent Michaelis-Menten constant decreases with increasing product concentration. Also, the parameters here are such that $A - K_m \gg \sqrt{C}$ and $[P] \gg K_{D,P}$, and it is observed that $K_m^{app} \sim K_{D,S}$.

The curves shown in Figure 5.8 illustrate the effect of product concentration on the product formation rate. It is seen that the higher the initial concentration of product is, the lower the maximal rate of product formation is, which is consistent with the formula (5.16) for sufficiently high $[S]$. Figure 5.9 shows the Lineweaver-Burk plots of the product formation rate formula 5.16 for $[P] = 0$ and one value with $[P] > 0$. Notice that the intersection point between the line for $[P] > 0$ and $1/v$-axis is higher than that between the line for $[P] = 0$ and $1/v$-axis. This implies that the maximal rate of product formation decreases when the initial concentration of product increases.

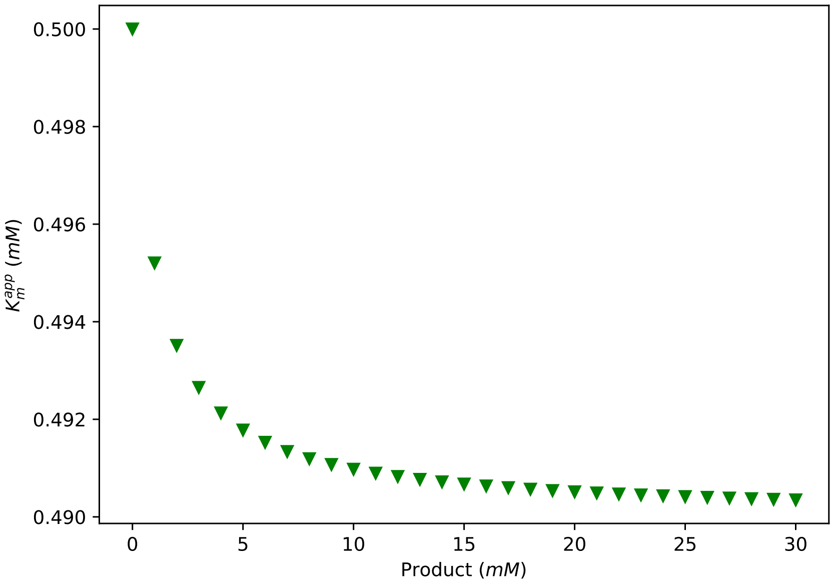


Figure 5.7: *Model II.* Plot of K_m^{app} as a function of $[P]$ as given by equation (5.18). The parameter used here are given by $e_0 = 0.1\ mM$, $k_0 = 10.0\ s^{-1}$, $k_1 = 1000.0\ mM^{-1}s^{-1}$, $k_{-1} = 490.0\ s^{-1}$, $k_2 = 1000.0\ mM^{-1}s^{-1}$, $k_{-2} = 100.0\ s^{-1}$, $K_m = 0.5\ mM$, $K_{D,P} = 0.1\ mM$, and $K_{D,S} = 0.49\ mM$.

5.2.1 Model II and glucose phosphorylation by mutant hexokinase I

The model we have just developed may be applicable to a mutant hexokinase I system. The mutant hexokinase I molecule in question is a hexokinase I enzyme molecule with the N binding site for ATP and the C binding site for $G6P$ deactivated; see Figure 5.10. This mutant is important because it has been successfully used in experimental studies [178, 180] to establish that the enzyme has two binding sites for $G6P$. We omit the technical details here - see [178, 180] for more information. The active site here is the binding site for ATP, so that ATP corresponds to the substrate species S in the model. The product P here is $G6P$ since the binding of $G6P$ to its N binding site allosterically inhibits the binding site of ATP. However, the correspondence between the model and the mutant hexokinase I system falls down here since $G6P$ is not a direct product of ATP binding. However, $G6P$ would be the effective product of ATP binding if ATP binding is the rate-limiting step for product formation. This would be the case for sufficiently high concentrations of glucose, for example. The model also does not take account of phosphate binding, and so would only apply to the mutant hexokinase I system if phosphate concentrations are sufficiently low. However, in these circemstances, the rate of production of $G6P$ may be approximated by (see (5.16)).

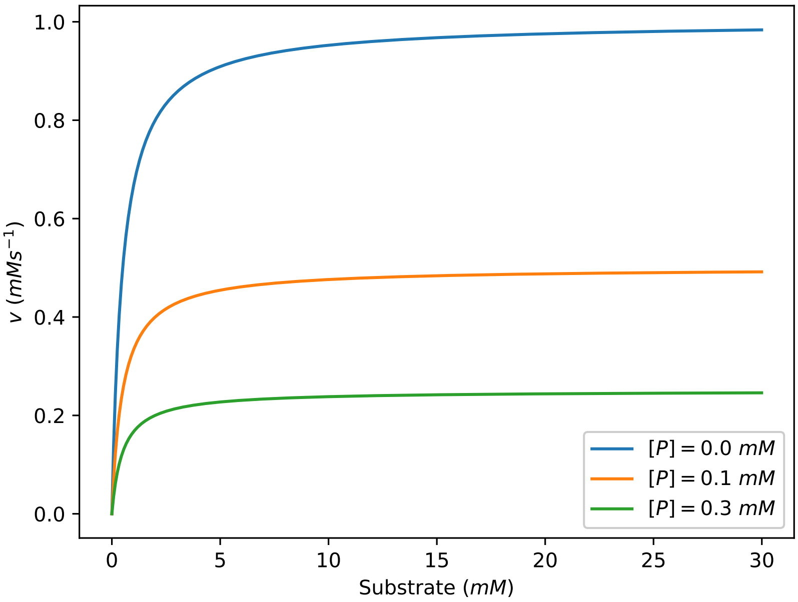


Figure 5.8: *Model II.* Plots of the product formation rate illustrating the effect of product concentration on the maximal rate of enzyme production (*Model II*). The parameter values used here are given by $e_0 = 0.1\ mM$, $k_0 = 10.0\ s^{-1}$, $k_1 = 1000.0\ mM^{-1}s^{-1}$, $k_{-1} = 490.0\ s^{-1}$, $k_2 = 1000.0\ mM^{-1}s^{-1}$, $k_{-2} = 100.0\ s^{-1}$, and $[P] = 0.0,\ 0.1,\ 0.3\ mM$.

$$v = \frac{V_{max}}{1 + [G6P]/K_{D,G6P}} \frac{[ATP]^2 + A[ATP]}{[ATP]^2 + B[ATP] + C}, \tag{5.19}$$

where

$$\begin{aligned} A &= K_{D,ATP} + (1 + [G6P]/K_{D,G6P})k_{-2}/k_1, \\ B &= K_{D,ATP} + K_m + (1 + [G6P]/K_{D,G6P})k_{-2}/k_1, \\ C &= K_{D,ATP}[K_m + (1 + [G6P]/K_{D,G6P})k_{-2}/k_1] + k_0k_{-2}/k_1^2, \end{aligned} \tag{5.20}$$

and

$$V_{max} = k_0e_0, \quad K_{D,ATP} = \frac{k_{-1}}{k_1}, \quad K_{D,G6P} = \frac{k_{-2}}{k_2}, \quad K_m = \frac{k_0 + k_{-1}}{k_1}.$$

5.3 Conclusions

The detailed mathematical model describing the phosphorylation of glucose by Hexokinase I presented in the previous chapter was complex, and combined three mechanisms for enzyme inhibition. Hence, it was impossible to obtain simple analytical expressions for the rate of product formation. It was also difficult to obtain simple qualitative insights into the enzyme behaviour. In an

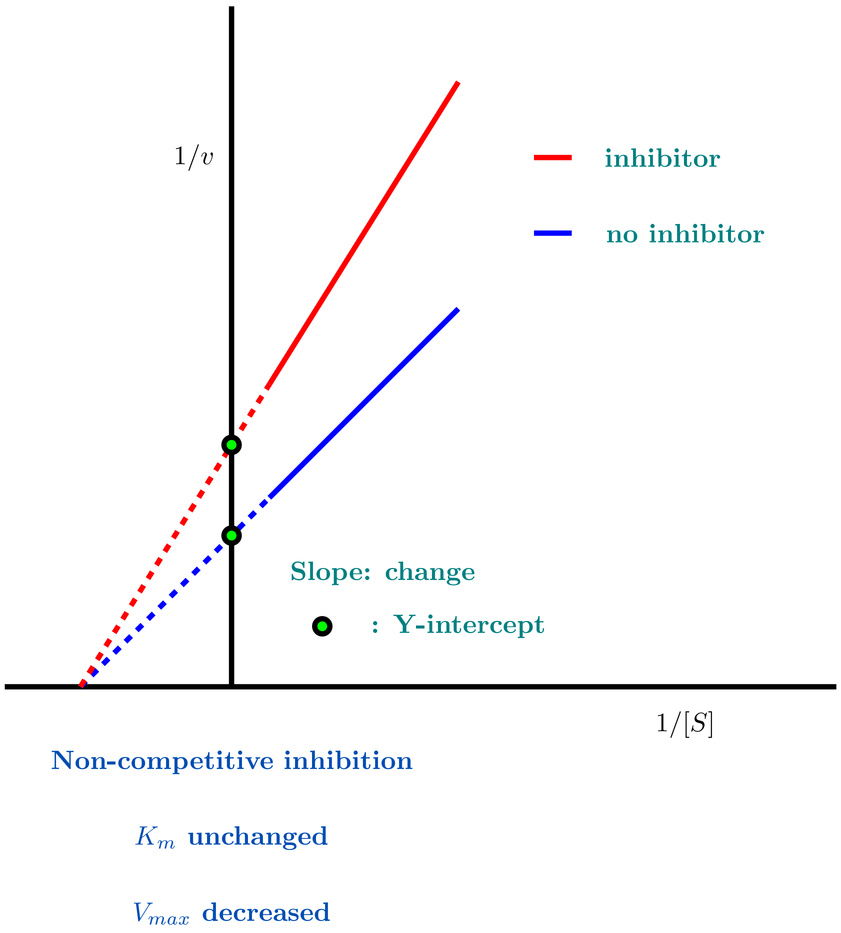


Figure 5.9: *Model II.* Lineweaver-Burk plots of the product formation rate formula (5.16) for $[P] = 0$ and $[P] > 0$.

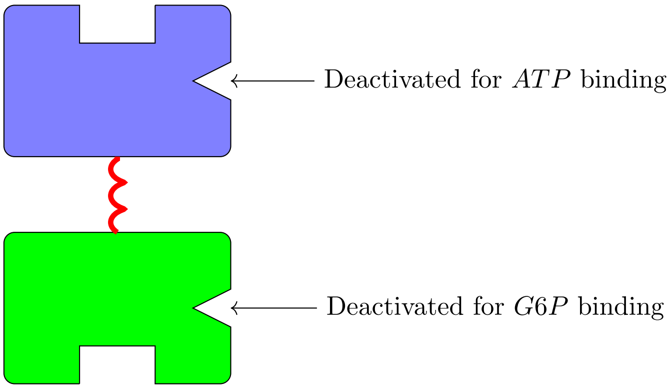


Figure 5.10: Mutant hexokinase I.

effort to overcome those deficiencies, we developed two simpler related models in this chapter. The first model considers the mechanism of competitive product inhibition only, while the second model concentrates on allosteric inhibition only. For each of these models, a formula of the rate of product formation was formulated. Also, an application of the first model to the phosphorylation of glucose was described. The second model was seen to model, under appropriate conditions, the phosphorylation of glucose by mutant hexokinase I. The second model may be applied to describe the reaction of horseradish peroxidase with a fluorogenic substrate [196].

Chapter 6

Discussion

In this thesis, we have formulated mathematical models describing the behaviour of some carbohydrate enzymes. We now summarise these models and indicate how they may be expanded and improved upon. We begin by considering the mathematical model for the degradation of hyaluronan by *streptococcus pneumoniae* hyaluronate lyase discussed in Chapter 3.

6.1 Modelling hyaluronan degradation by *streptococcus pneumoniae* hyaluronate lyase

6.1.1 The first detailed mathematical model

To the best of my knowledge, the model formulated is the first that describes in detail the degradation of hyaluronan by *streptococcus pneumoniae* hyaluronate lyase. It was developed based on what is currently known about hyaluronan and the kinetic mechanism of *streptococcus pneumoniae* hyaluronate lyase. Using available experimental data, the model parameters values were estimated, and it was found that the model results corresponded well with experiments. The model results were also found to be consistent with experimental data for other bacterial hyaluronidases. This suggests that the model may have wider applicability. It should also be noted that additional experimental data is required for complete model validation.

6.1.2 The model may be refined and expanded

The hyaluronan degradation model presented in Chapter 3 may be expanded upon and refined. For example, it is known that hyaluronan in solution may adopt secondary and tertiary structures [144, 145]. It is likely that these sructures may affect the accessibility of some of the glycosidic bonds for the enzyme. Such effects have implications for the parameters in our modelling

[146]. It is probable that some of the model parameters estimated are in fact effective parameters that implicitly incorporate effects not explicitly modelled. These issues could form the basis of future interesting studies that incorporate more of the mechanistic details of the degradation process.

Another issue that requires further experimental and theoretical investigation is enzyme inhibition. In the current study, we assume that enzyme activity remains constant throughout the degradation process. However, many enzymatic degradation processes for polysaccharides are known to be subject to various inhibitory processes [197, 198, 199, 200], and such effects may also play a role in the current context.

6.2 Modelling the phosphorylation of glucose by human hexokinase I

Here, we give some discussion of the glucose phosphorylation model presented in Chapter 4, as well as some indications as to how it can be further developed.

6.2.1 The mathematical model

This mathematical model is the first comprehensive model describing the phosphorylation of glucose by the enzyme hexokinase I. It was developed based on what is currently known about hexokinase I, and following a careful review of the relevant literature. Glucose phosphorylation is the first step of the glycolytic pathway, and so it is carefully regulated by cells. The regulation of hexokinase I is quite complex and includes three inhibitory mechanisms: a competitive product inhibitory mechanism, an allosteric inhibitory mechanism, and a competitive inhibitory mechanism. We used the model to help unpick the regulatory behaviour of hexokinase I. In particular, we obtained the following results.

- *Numerical simulations.* The model solutions obtained were consistent with the known behaviour of hexokinase I. For example, it was found that the rate of phosphorylation decreased with increasing concentration of $G6P$. Also, it was found that low phosphate concentrations antagonises hexokinase I inhibition, while high phosphate concentrations inhibit hexokinase I.

- *Global sensitivity analysis.* The results of this analysis indicate that the rate of phosphorylation is sensitive to the following factors: the turnover rate of the enzyme; the rate of of binding/unbinding of ATP to/from the C domain of the enzyme; the rate of binding/unbinding of $G6P$ to/from the C domain of the enzyme with a P_i molecule bound at the N

domain for low phosphate concentration; and the rate of phosphate binding/unbinding to/from the C domain of the enzyme for high phosphate concentration.

- *Simplified model.* One reduced model was developed based on the results of the sensitivity analysis. This simpler model produces results that closely match the results of the full model.

6.2.2 The model may be extended

Although the model we have developed for glucose phosphorylation is comprehensive and detailed, there is some scope for improvement. For example, the full detail of the Bi Bi mechanism [172] could be incorporated in the modelling. Also, glucose-6-phosphate binding to its N binding site not only allosterically inhibits the enzyme but also stimulates enzyme release from mitochondria [192, 193], and no attempt has been made to describe this release behaviour. Furthermore, the possible inhibition of hexokinase I by ADP [185] has not been explored in the current study. Also, some glycolytic intermediates can inhibit hexokinase I, examples being 2,3-diphosphoglycerate, glycerate-3-phosphate and fructose-1,6-diphosphate [185]. This inhibition has not been incorporated in the current modelling.

6.3 Modelling enzyme with product inhibition

6.3.1 Two mathematical models

In Chapter 5, we have developed two mathematical models (Model I and Model II) for the kinetics of enzymes with product inhibition. Model I describes enzymes subject to competitive inhibition, and Model II describes enzymes subject to allosteric inhibition. Model II is new, but Model I is not. Nevertheless, we used a careful scaling argument to carefully justify the formulation of Model I and clarified the assumptions made for its derivation. Formulae for the product formation rate of each model were developed.

6.3.2 Model applications

The models may be applied to a mini hexokinase I system and to a mutant hexokinase I system. Model I may model a mini hexokinase I system involving mini hexokinase I, glucose (at sufficiently high concentration), and ATP. A mini hexokinase I enzyme molecule contains the C domain of the hexokinase I molecule only. Model II may describe a system involving mutant hexokinase I, glucose (at sufficiently high concentration), and ATP. In this case, the mutant hexokinase I enzyme molecule is a hexokinase I enzyme molecule with the N binding site for ATP deactivated, and the C binding site for $G6P$ deactivated. It should be noted that in order for a $G6P$ molecule to be formed, a glucose molecule and an ATP molecule must be bound at their C binding site. However, $G6P$ would be the effective product of ATP binding if ATP binding is the rate-limiting step for product formation. This would be the case for sufficiently high concentrations of glucose, for example.

6.4 Some other ideas for future research

The following are some suggestions for future work on modelling carbohydrate enzymes.

6.4.1 Modelling the cellular synthesis of hyaluronan [201]

It is now known that hyaluronan is involved in cancer metastasis [202, 203]. It is also speculated that the length of hyaluronan chains may play a vital important role in cancer resistance. The remarkable longevity of naked mole rats may be due to the fact that they secrete extremely high-molecular-weight hyaluronan, over five times larger than human or mouse hyaluronan [204]. Hence, it would be of interest to develop a mathematical model describing the cellular synthesis of hyaluronan to help identify the key factors involved in this process, and in particular, to help pinpoint those factors affecting ultimate chain length.

6.4.2 Modelling the behaviour of a bisubstrate enzyme with competitive product inhibition

This system involves a bisubstrate enzyme E, substrates S_1, S_2, and products P_1, P_2. The kinetic mechanism of the enzyme is a Bi Bi random mechanism, and the product P_1 inhibits the enzyme by competing with the substrate S_2. A schematic diagram of the system reactions is depicted in Figure 6.1. This model is applicable to a mini hexokinase I system involving mini hexokinase I enzyme, glucose, ATP, glucose-6-phosphate, and ADP. The correspondence between the model and this system is as follows:

- mini hexokinase I is the enzyme E;
- glucose and ATP correspond to the substrates S_1 and S_2, respectively;
- glucose-6-phosphate and ADP correspond to the products P_1 and P_2, respectively.

Recall that glucose-6-phosphate competes with ATP for the C binding site on a mini hexokinase I molecule.

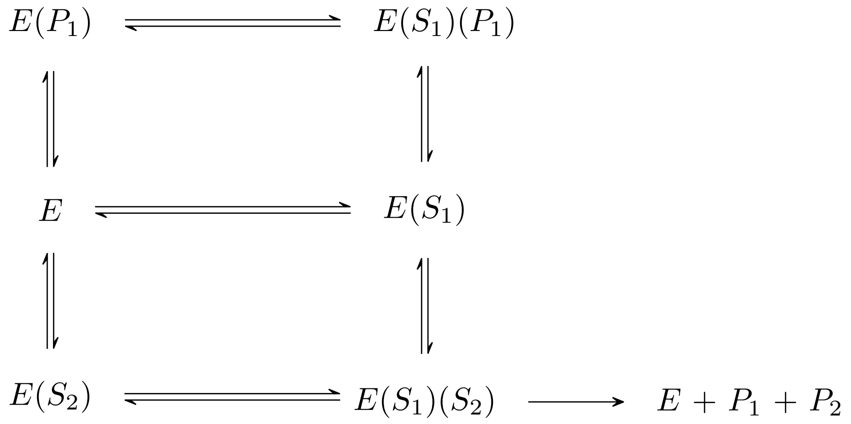


Figure 6.1: A schematic diagram for the reactions of a bisubstrate enzyme system with competitive product inhibition. E denotes the enzyme, S_1, S_2 the substrates, P_1, P_2 the products, and $E(S_1)$, $E(S_2)$, $E(S_1)(S_2)$, $E(P_1)$, and $E(S_1)(P_1)$ the enzyme complexes. The product P_1 inhibits the enzyme by competing with the S_2 substrate.

6.4.3 Modelling a bisubstrate enzyme with allosteric product inhibition

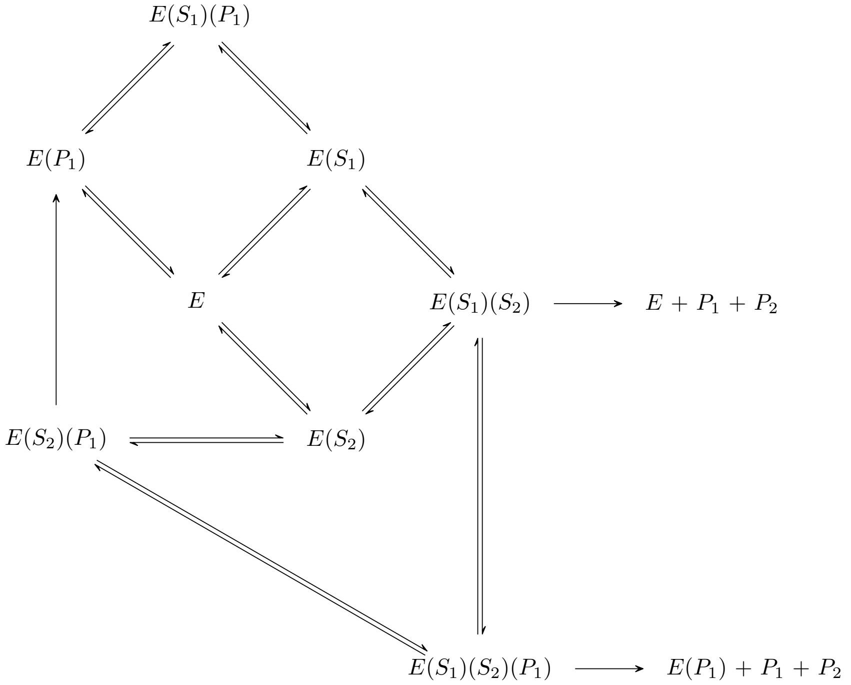


Figure 6.2: A schematic diagram for the reactions of a bisubstrate enzyme system with allosteric product inhibition. E denotes the enzyme, S_1, S_2 the substrates, P_1, P_2 the products, and $E(S_1)$, $E(S_2)$, $E(S_1)(S_2)$, $E(P_1)$, $E(S_1)(P_1)$, $E(S_2)(P_1)$, and $E(S_1)(S_2)(P_1)$ the enzyme complexes. The binding of a product P_1 molecule to an enzyme molecule makes a conformational change to the binding site for the S_2 substrate so that S_2 cannot then bind to the enzyme molecule.

This system consists of a bisubstrate enzyme E, substrates S_1, S_2, and products P_1, P_2. The kinetic mechanism of the enzyme is a Bi Bi random mechanism, and the product P_1 inhibits the enzyme E by binding to a distinct site of the enzyme. The binding of a P_1 molecule to an enzyme molecule results in a conformational change of the binding site for the S_2 substrate so that S_2 cannot then bind to the enzyme molecule. A schematic diagram for the system of reactions is depicted in Figure 6.2. One of the principal aims of this work would be to derive a rate equation for the system.

The mechanism just described is of interest because it may be applicable to the enzyme horseradish peroxidase. Horseradish peroxidase is a well-known enzyme that is widely used in the food industry and in experimental studies [205]. This enzyme has a Bi Bi random kinetic mechanism, and it is allosterically inhibited by its product [196]. The correspondence between the model and this system is as follows:

- horseradish peroxidase is the enzyme E;
- H_2O_2 and Amplex Red reagent correspond to the substrates S_1 and S_2, respectively [206];
- oxidised Amplex Red reagent and H_2O correspond to the products P_1 and P_2, respectively.

Appendices

Appendix A

Mathematical model and Computational programs for Chapter 3

A.1 Mathematical model

Here we list the complete set of governing equations for the model in Chapter 3. The notation used here is explained in the nomenclature at the beginning of the chapter. Information on how these equations are constructed can be found in Section 3.2.2 and in Figure 3.6.

$$\begin{aligned}
\frac{d[D_1]}{dt} &= -k_{ads}[E].[D_1] + k_{trans}[E \circ D_1] + k_{clv}\sum_{i=2}^{N}[E \diamond D_i], \\
\frac{d[D_2]}{dt} &= -2k_{ads}[E].[D_2] + k_{des}[E \diamond D_2] + k_{clv}\sum_{i=3}^{N}\frac{1}{i-2}[E \times D_i], \\
\frac{d[D_3]}{dt} &= -3k_{ads}[E].[D_3] + k_{des}[E \diamond D_3] + k_{des}[E \times D_3] \\
&\quad +k_{clv}\sum_{i=4}^{N}\frac{1}{i-2}[E \times D_i], \\
&\vdots \\
\frac{d[D_{N-1}]}{dt} &= -(N-1)k_{ads}[E].[D_{N-1}] + k_{des}[E \diamond D_{N-1}] \\
&\quad +k_{des}[E \times D_{N-1}] + \frac{1}{N-2}k_{clv}[E \times D_N], \\
\frac{d[D_N]}{dt} &= -Nk_{ads}[E].[D_N] + k_{des}[E \diamond D_N] + k_{des}[E \times D_N],
\end{aligned}$$

$$
\begin{aligned}
\frac{d[E]}{dt} &= -k_{ads}[E]\left(\sum_{i=1}^{N} i[D_i]\right) + k_{trans}[E\circ D_1] \\
&\quad +k_{des}\left(\sum_{i=2}^{N}[E\diamond D_i] + \sum_{i=3}^{N}[E\times D_i]\right), \\
\frac{d[E\circ D_1]}{dt} &= -k_{trans}[E\circ D_1] + k_{ads}[E].[D_1] + k_{clv}[E\diamond D_2] \\
&\quad +k_{clv}\sum_{i=3}^{N}\frac{1}{i-2}[E\times D_i], \\
\frac{d[E\circ D_k]}{dt} &= -k_{trans}[E\circ D_k] + k_{ads}[E].[D_k] + k_{clv}[E\diamond D_{k+1}] \\
&\quad +k_{revtr}[E\diamond D_k] + k_{clv}\sum_{m=k+2}^{N}\frac{1}{m-2}[E\times D_m], \\
&\quad 2\le k\le N-2, \\
\frac{d[E\circ D_{N-1}]}{dt} &= -k_{trans}[E\circ D_{N-1}] + k_{ads}[E].[D_{N-1}] + k_{clv}[E\diamond D_N] \\
&\quad +k_{revtr}[E\diamond D_{N-1}] \\
\frac{d[E\circ D_N]}{dt} &= -k_{trans}[E\circ D_N] + k_{ads}[E].[D_N] + k_{revtr}[E\diamond D_N]. \\
\frac{d[E\diamond D_j]}{dt} &= -(k_{des}+k_{clv}+k_{revtr})[E\diamond D_j] + k_{trans}[E\circ D_j] \\
&\quad +k_{ads}[E].[D_j],\ 2\le j\le N, \\
\frac{d[E\times D_i]}{dt} &= -(k_{des}+k_{clv})[E\times D_i] + (i-2)k_{ads}[E].[D_i],\ 3\le i\le N.
\end{aligned}
$$

These equations are solved subject to the initial conditions

$$
\begin{aligned}
&[E](t=0)=E_0, && \\
&[D_N](t=0)=D_0, && \\
&[D_i](t=0)=0, && 1\le i\le N-1, \\
&[E\times D_i](t=0)=0, && 3\le i\le N, \\
&[E\diamond D_i](t=0)=0, && 2\le i\le N. \\
&[E\circ D_i](t=0)=0, && 1\le i\le N,
\end{aligned}
$$

where E_0, D_0 give the initial concentrations of enzyme and polymer chains of degree N, respectively.

A.2 Computational programs

See the NEXT PAGE.

```
# Supplementary material
#
# The program used to estimate parameters and
# calculate sensitivity indices. It can be
# executed with the following libraries
# in Python 2.7 or later!
#
# Libraries for parameter estimation
import numpy as np
from scipy.integrate import odeint
from scipy import integrate
from scipy.optimize import minimize
#
# Library for plotting
import matplotlib.pyplot as plt
#
#
#
#=================================================
# n: number of disaccharides of a HA chain.
# Take care when choosing n for
# sensitivity analysis
n = 125
#
#=================================================
# Create solution to the model that is a
# vector-value function of p, initial_cond, t0,
# t_end and stpz, where p, initial_cond,
# t0, t_end and stpz are a parameter vector, a
# vector of initial concentrations, starting time,
# ending time, and step size of time
# interval, respectively.
#
def sol(p,initial_cond,t0,t_end,stpz):
    # Parameters
    k1, k2, k3, k5, k6 = p
    # time-grid-----
    t = np.arange(t0, t_end, stpz)
    # Model
    def funct(y,t):
        # if n = 4,
        # y[0] = E
        # y[i] = D[i], i=1,n
        # y[i] = EoD[i-n], i=n+1,2n, 5,6,7,8
        # y[i] = EvD[i-2n+1], i=2n+1,3n-1, 9,10,11
        # y[i] = ExD[i-3n+3], i=3n,4n-3, 12,13
        #n = 20
        #k = 4*n - 2
        # the model equations
            # sum of D[i], i=1,n
        #sumD = sum(y[i] for i in range(1,n+1))
            # sum of i*D[i], i=1,n
        #sumiD = sum(i*y[i] for i in range(1,n+1))
            # sum of EoD[i], i=1,n
        #sumC = sum(y[i] for i in range(n+1,2*n+1))
            # sum of EvD[i], i=2,n
        #sumT = sum(y[i] for i in range(2*n+1,3*n))
            # sum of ExD[i], i=3,n
        #sumX = sum(y[i] for i in range(3*n,4*n-2)
        #===============================
        f = []
        # Eq. for Enzyme = y[0]
        f.append(-k1*y[0]*sum(i*y[i] for i in range(1,n+1)) + k5*y[n+1]
            + k2*(sum(y[i] for i in range(2*n+1,3*n))
            + sum(y[i] for i in range(3*n,4*n-2))))
        # Eq. for D[1] = y[1]
        f.append(-k1*y[0]*y[1] + k5*y[n+1]
            + k3*sum(y[i] for i in range(2*n+1,3*n)))
        # Eq. for D[2] = y[2]
        f.append(-k1*2*y[0]*y[2] + k2*y[2*n+1]
            + k3*sum(y[i]/(i-3*n+1) for i in range(3*n,4*n-2)))
        # Eqs. for D[i] = y[i], i=3,n-1
        for i in range(3,n):
            f.append(-k1*i*y[0]*y[i] + k2*y[2*n+i-1] + k2*y[3*n+i-3]
                + k3*sum(y[k]/(k-3*n+1) for k in range(3*n+1+i-3,4*n-2)))
        # Eq. for D[n] = y[n]
        f.append(-k1*n*y[0]*y[n] + k2*y[3*n-1] + k2*y[4*n-3])
        # Eq. of EoD[1] = y[n+1]
        f.append(-k5*y[n+1] + k1*y[0]*y[1] + k3*y[2*n+1]
            + k3*sum(y[i]/(i-3*n+1) for i in range(3*n,4*n-2)))
        # Eqs. for EoD[2] = y[n+2], 2<=i<=n-2
        for i in range(n+2,2*n-1):
            f.append(-k5*y[i] + k1*y[0]*y[i-n] + k3*y[i+n] + k6*y[i+n-1]
                + k3*sum(y[k]/(k-3*n+1) for k in range(i+2*n-1,4*n-2)))
        # Eq. for EoD[n-1] = y[2*n-1]
        f.append(-k5*y[2*n-1] + k1*y[0]*y[n-1] + k3*y[3*n-1] + k6*y[3*n-2])
        # Eq. for EoD[n] = y[2*n]
        f.append(-k5*y[2*n] + k1*y[0]*y[n] + k6*y[3*n-1])
        # Eqs. for EvD[i] = y[2*n+i-1], i=2,n
        for i in range(2*n+1,3*n):
            f.append(-(k2 + k3 + k6)*y[i] + k5*y[i-n+1] + k1*y[0]*y[i-2*n+1])
        # Eqs. for ExD[i] = y[3n+i-3], i=3,n
        for i in range(3*n,4*n-2):
            f.append(-(k2 + k3)*y[i] +k1*(i-3*n+1)*y[0]*y[i-3*n+3])
        return(f)
    #================================
    # integrate the system---
    ds = integrate.odeint(funct,initial_cond,t)
    return(ds)
#
#================================
# Section for Parameter estimation
# Molecular weight of a disaccharide unit
#
b = 401.30
#
# Initial conditions
#
y0 = []
E0 = 1.21e-5
D0 = 2.64e-4/26**2      # 2.64e-2/n (real)
y0.append(E0)
for i in range(1,75):
    y0.append(0.0)
for i in range(75,101):
    y0.append((i-74)*D0)
for i in range(101,n+1):
    y0.append((n+1-i)*D0)
for i in range(n+1,4*n-2):
    y0.append(0.0)
y0
#
#================================
# Time grid 1-----
stpz = 1e-4
t0 = 0.0
t_end = 48.0 + stpz
t1 = np.arange(t0, t_end, stpz)
#
#================================
# Original guesses
#
p0 = [9998.23, 109.36, 2695.43, 2096.69, 4.70]
#
#================================
# Data section
    # Data
    # Time grid
```

```
Td = [0.0, 0.3528, 0.9119, 1.8932, 3.9141, 6.0, 24.0, 48.0]
        # Pneumococcal
Zd = [180.1559*D0, 0.6969, 1.3391, 2.7056, 4.5154, 4.6042, 4.6947, 4.7562]
indices = [int(x*1e+4) for x in Td]
#
#================================
# Section for parameter estimation
    # Score fit of the model to data
def score(p):
    # Solutions
    y = sol(p, y0, 0.0, t_end, stpz)
    # Reducing ends
    r_ends = 180.1559*sum(y[:,i] for i in range(1,4*n-2))
    Zm = np.take(r_ends, indices)
    ss = sum((x - y)**2 for x, y in zip(Zm, Zd))
    return(ss)
#
#================================
# Minimize the score
#
# Original bounds
#
#bnds = ((5000,3e+4), (0,1e+3), (500,5e+3), (500,5e+3), (0,1e+2))
#
s_fit = minimize(score, p0, method='nelder-mead', options={'xtol':1e-4, 'disp':
True})
new_p = s_fit.x
print('for n =' + str(n))
print(new_p)
#
#---------------------------------------------------------------------------------
# Section for plotting
# the model curve and the data
#
if 1 == 10:
        y_sol = sol(p0, y0, t0, t_end, stpz)
    #
        # Plot Reducing sugars
    fg1 = plt.figure(1)
    plt.plot(t1, 180.1559*sum(y_sol[:,i] for i in range(1,4*n-2)))
    plt.plot(Td, Zd, 'ro')
    plt.legend(['Model curve', 'Data'])
    plt.xlabel('TIME (HOURS)')
    plt.ylabel('INCREASE IN REDUCING ENDS' + '\n' + 'AS GLUCOSE $mg/ml$')
    #fg1.savefig('data.eps')
    plt.show()
```

```
# Global sensitivity analysis program
#
#
# Libraries needed
from SALib.sample import saltelli
from SALib.analyze import sobol
import numpy as np
from scipy.integrate import odeint
from scipy import integrate
#
#==================================================
# No. of disacch.
# n = 35 and 40 chosen to implement with.
n = 35
#
#==================================================
# Solutions to model
def sol(p,initial_cond,t0,t_end,stpz):
    # Parameters
    k1, k2, k3, k5, k6 = p
    # time-grid-----
    t = np.arange(t0, t_end, stpz)
    # Model
    def funct(y,t):
        # if n = 4,
        # y[0] = E
        # y[i] = D[i], i=1,n
        # y[i] = EoD[i-n], i=n+1,2n, 5,6,7,8
        # y[i] = EvD[i-2n+1], i=2n+1,3n-1, 9,10,11
        # y[i] = ExD[i-3n+3], i=3n,4n-3, 12,13
        #n = 20
        #k = 4*n - 2
        # the model equations
            # sum of D[i], i=1,n
        #sumD = sum(y[i] for i in range(1,n+1))
            # sum of i*D[i], i=1,n
        #sumiD = sum(i*y[i] for i in range(1,n+1))
            # sum of EoD[i], i=1,n
        #sumC = sum(y[i] for i in range(n+1,2*n+1))
            # sum of EvD[i], i=2,n
        #sumT = sum(y[i] for i in range(2*n+1,3*n))
            # sum of ExD[i], i=3,n
        #sumX = sum(y[i] for i in range(3*n,4*n-2)
        #================================
        f = []
        # Eq. of Enzyme = y[0]
        f.append(-k1*y[0]*sum(i*y[i] for i in range(1,n+1)) + k5*y[n+1]
            + k2*(sum(y[i] for i in range(2*n+1,3*n))
            + sum(y[i] for i in range(3*n,4*n-2))))
        # Eq. of D[1] = y[1]
        f.append(-k1*y[0]*y[1] + k5*y[n+1]
            + k3*sum(y[i] for i in range(2*n+1,3*n)))
        # Eq. of D[2] = y[2]
        f.append(-k1*2*y[0]*y[2] + k2*y[2*n+1]
            + k3*sum(y[i]/(i-3*n+1) for i in range(3*n,4*n-2)))
        # Eqs. of D[i] = y[i], i=3,n-1
        for i in range(3,n):
            f.append(-k1*i*y[0]*y[i] + k2*y[2*n+i-1] + k2*y[3*n+i-3]
                + k3*sum(y[k]/(k-3*n+1) for k in range(3*n+1+i-3,4*n-2)))
        # Eq. of D[n] = y[n]
        f.append(-k1*n*y[0]*y[n] + k2*y[3*n-1] + k2*y[4*n-3])
        # Eq. of EoD[1] = y[n+1]
        f.append(-k5*y[n+1] + k1*y[0]*y[1] + k3*y[2*n+1]
            + k3*sum(y[i]/(i-3*n+1) for i in range(3*n,4*n-2)))
        # Eqs. of EoD[2] = y[n+2], 2<=i<=n-2
        for i in range(n+2,2*n-1):
            f.append(-k5*y[i] + k1*y[0]*y[i-n] + k3*y[i+n] + k6*y[i+n-1]
                + k3*sum(y[k]/(k-3*n+1) for k in range(i+2*n-1,4*n-2)))
        # Eq. of EoD[n-1] = y[2*n-1]
        f.append(-k5*y[2*n-1] + k1*y[0]*y[n-1] + k3*y[3*n-1] + k6*y[3*n-2])
        # Eq. of EoD[n] = y[2*n]
        f.append(-k5*y[2*n] + k1*y[0]*y[n] + k6*y[3*n-1])
        # Eqs. of EvD[i] = y[2*n+i-1], i=2,n
        for i in range(2*n+1,3*n):
            f.append(-(k2 + k3 + k6)*y[i] + k5*y[i-n+1] + k1*y[0]*y[i-2*n+1])
        # Eqs. of ExD[i] = y[3n+i-3], i=3,n
        for i in range(3*n,4*n-2):
            f.append(-(k2 + k3)*y[i] +k1*(i-3*n+1)*y[0]*y[i-3*n+3])
        return(f)
    #===============================
    # integrate the system---
    ds = integrate.odeint(funct,initial_cond,t)
    return(ds)
#
#===============================
#====================================================
# Model for Sensitivity Analysis
def model(p):
    #===============================
    # initial conditions
    y0 = []
    E0 = 1.21e-5 # 0.00001204 mmol/ml
    D0 = 2.64e-2/n # 0.02491957867/n
    y0.append(E0)
    for i in range(1,n):
        y0.append(0.0)
    y0.append(D0)
    for i in range(n+1,4*n-2):
        y0.append(0.0)
    y0
    #==============================
    # Solutions
    y = sol(p,y0,0.0,6.0 + 0.02, 0.01)
    # Disaccharide in mg/ml
    F = 401.3*y[:,1]
    return(F)
    #
#=================================================
# Define problem of sensitivity analysis
problem = {
    'num_vars': 5,
    'names': ['k1', 'k2', 'k3', 'k5', 'k6'],
    'bounds': [[0.9*9998.23,1.1*9998.23 ], [0.9*109.36,1.1*109.36],
[0.9*2695.43,1.1*2695.43], [0.9*2096.69,1.1*2096.69], [0.9*4.7,1.1*4.7]]
}
#
# Generate samples
param_values = saltelli.sample(problem, 1000)
#
#=================================================
# Number of model points at which implementing SA
print('For n = ' + str(n) + '\n')
L = [101, 201, 301, 451, 601]
for i in L:
    # Run model
    Y = np.zeros([param_values.shape[0]])
    for j, X in enumerate(param_values):
        K = model(X)
        Y[j] = K[i]
    # Perform analysis
    Si = sobol.analyze(problem, Y, print_to_console=True)
    #
    # Print the first-order sensitivity indices
    print(' and for i = ' + str(i) + '\n')
    print(Si['S1'])
    # Print the total-order sensitivity indices
    print(Si['ST'])
#=========================================================
```

Appendix B

Mathematical model and Computational programs for Chapter 4

B.1 The chemical reactions

Here we list all of the enzymatic reactions included in the mathematical model. For brevity, we introduce the following notation.

E:	free hexokinase I enzyme,	0:	glucose,	1:	ATP,
2:	$G6P$,			3:	P_i,	4:	ADP,

k_0 : catalytic constant rate,

k_1, k_3, k_5, k_7 : adsorption constant rates of glucose, ATP, $G6P$, and P_i on the N binding sites, respectively.

k_{-1}, k_{-3}, k_{-5}, k_{-7} : desorption constant rates of glucose, ATP, $G6P$, and P_i from the N binding sites, respectively.

k_2, k_4, k_6, k_8 : adsorption constant rates of glucose, ATP, $G6P$, and P_i on the C binding sites, respectively.

k_{-2}, k_{-4}, k_{-6}, k_{-8} : desorption constant rates of glucose, ATP, $G6P$, and P_i from the C binding sites, respectively.

k_9, k_{-9} : adsorption and desorption constant rates of $G6P$ on and from the Cbinding site of complexes of enzyme with one P_i molecule bound at the N binding site.

E^x_{yz} : an enzyme complex with an x molecule bound at its N binding site, a y molecule bound at its C site, and a z molecule bound at its C site.

$$E+0 \xrightleftharpoons[k_{-1}]{k_1} E^0, \qquad E+0 \xrightleftharpoons[k_{-2}]{k_2} E_0, \qquad E+1 \xrightleftharpoons[k_{-3}]{k_3} E^1, \qquad E+1 \xrightleftharpoons[k_{-4}]{k_4} E_1,$$

$$E+2 \xrightleftharpoons[k_{-5}]{k_5} E^2, \qquad E+2 \xrightleftharpoons[k_{-6}]{k_6} E_2, \qquad E+3 \xrightleftharpoons[k_{-7}]{k_7} E^3, \qquad E+3 \xrightleftharpoons[k_{-8}]{k_8} E^0,$$

$$\boxed{E^0}+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^0_0,\quad E^0+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^{01},\quad E^0+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^0_1,\quad E^0+2\underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}}E^{02},$$

$$E^0+2\underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}}E^0_2,\quad E^0+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^{03},\quad E^0+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^0_3,\quad \boxed{E_0}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^0_0,$$

$$E^0+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^{01},\quad E^0+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^0_1,\quad E^0+2\underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}}E^{02},\quad E^0+2\underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}}E^0_2,$$

$$E^0+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^{03},\quad E^0+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^0_3,\quad \boxed{E^1}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{01},\quad E^1+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^1_0,$$

$$E^1+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^1_1,\quad E^1+2\underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}}E^1_2,\quad E^1+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^1_3,\quad \boxed{E_1}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^0_1,$$

$$E_1+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E_{01},\quad E_1+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^1_1,\quad E_1+2\underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}}E^2_1,\quad E_1+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^3_1,$$

$$\boxed{E^2}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{02},\quad E^2+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^2_0,\quad \boxed{E_2}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^0_2,\quad E_2+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E_{02},$$

$$E_2+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^1_2,\quad E_2+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^3_2,\quad \boxed{E^3}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{03},\quad E^3+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^3_0,$$

$$E^3+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^3_1,\quad E^3+2\underset{k_{-9}}{\overset{k_9}{\rightleftharpoons}}E^3_2,\quad E^3+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^3_3,\quad \boxed{E_3}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^0_3,$$

$$E_3+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E_{03},\quad E_3+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^1_3,\quad E_3+2\underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}}E^2_3,\quad E_3+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^3_3,$$

$$\boxed{E^{01}}+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^{01}_0,\quad E^{01}+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^{01}_1,\quad E^{01}+2\underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}}E^{01}_2,\quad E^{01}+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^{01}_3,$$

$$\boxed{E^{02}}+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^{02}_0,\quad \boxed{E^{03}}+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^{03}_0,\quad E^{03}+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^{03}_1,\quad E^{03}+2\underset{k_{-9}}{\overset{k_9}{\rightleftharpoons}}E^{03}_2,$$

$$E^{03}+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^{03}_3,\quad \boxed{E^0_0}+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^{01}_0,\quad E^0_0+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^0_{01},\quad E^0_0+2\underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}}E^{02}_0,$$

$$E^0_0+2\underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}}E^0_{02},\quad E^0_0+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^{03}_0,\quad E^0_0+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^0_{03},\quad \boxed{E^0_1}+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^0_{01},$$

$$E^0_1+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^{01}_1,\quad E^0_1+2\underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}}E^{02}_1,\quad E^0_1+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^{03}_1,\quad \boxed{E^0_2}+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^0_{02},$$

$$E^0_2+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^{01}_2,\quad E^0_2+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^{03}_2,\quad \boxed{E^0_3}+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^0_{03},\quad E^0_3+1\underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}}E^{01}_3,$$

$$E^0_3+2\underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}}E^{02}_3,\quad E^0_3+3\underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}}E^{03}_3,\quad \boxed{E^1_0}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{01}_0,\quad E^1_0+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^1_{01},$$

$$E^1_0+2\underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}}E^1_{02},\quad E^1_0+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^1_{03},\quad \boxed{E^1_1}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{01}_1,\quad E^1_1+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^1_{01},$$

$$\boxed{E^1_2}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{01}_2,\quad E^1_2+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^1_{02},\quad \boxed{E^1_3}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{01}_3,\quad E^1_3+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^1_{03},$$

$$\boxed{E^2_0}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{02}_0,\quad \boxed{E^2_1}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{02}_1,\quad E^2_1+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^2_{01},\quad \boxed{E^2_3}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{02}_3,$$

$$E^2_3+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^2_{03},\quad \boxed{E^3_0}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{03}_0,\quad E^3_0+1\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}}E^3_{01},\quad E^3_0+2\underset{k_{-9}}{\overset{k_9}{\rightleftharpoons}}E^3_{02},$$

$$E^3_0+3\underset{k_{-8}}{\overset{k_8}{\rightleftharpoons}}E^3_{03},\quad \boxed{E^3_1}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{03}_1,\quad E^3_1+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^3_{01},\quad \boxed{E^3_2}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{03}_2,$$

$$E^3_2+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^3_{02},\quad \boxed{E^3_3}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^{03}_3,\quad E^3_3+0\underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}}E^3_{03},\quad \boxed{E_{01}}+0\underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}}E^0_{01},$$

$$
\begin{array}{llll}
E_{01}+1 \xrightleftharpoons[k_{-3}]{k_3} E_{01}^{1}, & E_{01}+2 \xrightleftharpoons[k_{-5}]{k_5} E_{01}^{2}, & E_{01}+3 \xrightleftharpoons[k_{-7}]{k_7} E_{01}^{3}, & \boxed{E_{02}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{02}^{0}, \\
E_{02}+1 \xrightleftharpoons[k_{-3}]{k_3} E_{02}^{1}, & E_{02}+3 \xrightleftharpoons[k_{-7}]{k_7} E_{02}^{2}, & \boxed{E_{03}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{03}^{0}, & E_{03}+1 \xrightleftharpoons[k_{-3}]{k_3} E_{03}^{1}, \\
E_{03}+2 \xrightleftharpoons[k_{-5}]{k_5} E_{03}^{2}, & E_{03}+3 \xrightleftharpoons[k_{-7}]{k_7} E_{03}^{3}, & \boxed{E_{0}^{01}}+1 \xrightleftharpoons[k_{-4}]{k_4} E_{01}^{01}, & E_{0}^{01}+2 \xrightleftharpoons[k_{-6}]{k_6} E_{02}^{01}, \\
E_{0}^{01}+3 \xrightleftharpoons[k_{-8}]{k_8} E_{03}^{01}, & \boxed{E_{1}^{01}}+0 \xrightleftharpoons[k_{-2}]{k_2} E_{01}^{01}, & \boxed{E_{2}^{01}}+0 \xrightleftharpoons[k_{-2}]{k_2} E_{02}^{01}, & \boxed{E_{3}^{01}}+0 \xrightleftharpoons[k_{-2}]{k_2} E_{03}^{01}, \\
\boxed{E_{1}^{02}}+0 \xrightleftharpoons[k_{-2}]{k_2} E_{01}^{02}, & \boxed{E_{3}^{02}}+0 \xrightleftharpoons[k_{-2}]{k_2} E_{03}^{02}, & \boxed{E_{0}^{03}}+1 \xrightleftharpoons[k_{-4}]{k_4} E_{01}^{03}, & E_{0}^{03}+2 \xrightleftharpoons[k_{-9}]{k_9} E_{02}^{03}, \\
E_{0}^{03}+3 \xrightleftharpoons[k_{-8}]{k_8} E_{03}^{03}, & \boxed{E_{1}^{03}}+0 \xrightleftharpoons[k_{-2}]{k_2} E_{01}^{03}, & \boxed{E_{2}^{03}}+0 \xrightleftharpoons[k_{-2}]{k_2} E_{02}^{03}, & \boxed{E_{3}^{03}}+0 \xrightleftharpoons[k_{-2}]{k_2} E_{03}^{03}, \\
\boxed{E_{01}^{0}}+1 \xrightleftharpoons[k_{-3}]{k_3} E_{01}^{01}, & E_{01}^{0}+2 \xrightleftharpoons[k_{-5}]{k_5} E_{01}^{02}, & E_{01}^{0}+3 \xrightleftharpoons[k_{-7}]{k_7} E_{01}^{03}, & \boxed{E_{01}^{1}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{01}^{01}, \\
\boxed{E_{01}^{2}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{01}^{02}, & \boxed{E_{01}^{3}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{01}^{03}, & \boxed{E_{02}^{0}}+1 \xrightleftharpoons[k_{-3}]{k_3} E_{02}^{01}, & E_{02}^{0}+3 \xrightleftharpoons[k_{-7}]{k_7} E_{02}^{03}, \\
\boxed{E_{02}^{1}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{02}^{01}, & \boxed{E_{02}^{3}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{02}^{03}, & \boxed{E_{03}^{0}}+1 \xrightleftharpoons[k_{-3}]{k_3} E_{03}^{01}, & E_{03}^{0}+2 \xrightleftharpoons[k_{-5}]{k_5} E_{03}^{02}, \\
\boxed{E_{03}^{1}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{03}^{01}, & \boxed{E_{03}^{2}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{03}^{02}, & \boxed{E_{03}^{3}}+0 \xrightleftharpoons[k_{-1}]{k_1} E_{03}^{03}. &
\end{array}
$$

Note no substrate can bind to the $\boxed{E_0^{02}}$ complex. Here are all reactions that produce product

$$
\begin{aligned}
&E_{01} \xrightarrow{k_0} E+2+4, && E_{01}^{0} \xrightarrow{k_0} E^{0}+2+4, \\
&E_{01}^{1} \xrightarrow{k_0} E^{1}+2+4, && E_{01}^{2} \xrightarrow{k_0} E^{2}+2+4, \\
&E_{01}^{3} \xrightarrow{k_0} E^{3}+2+4, && E_{01}^{01} \xrightarrow{k_0} E^{01}+2+4, \\
&E_{01}^{02} \xrightarrow{k_0} E^{02}+2+4, && E_{01}^{03} \xrightarrow{k_0} E^{03}+2+4,
\end{aligned}
$$

B.2 The model equations

We now list the complete set of governing equations for the model. The notation used is explained in the chapter. Information on how these equations are constructed can also be found in the chapter.

$$
\begin{aligned}
\frac{d[E]}{dt} = {} & k_0[E_{01}] + k_{-1}[E^0] + k_{-2}[E_0] + k_{-3}[E^1] + k_{-4}[E_1] + k_{-5}[E^2] \\
& + k_{-6}[E_2] + k_{-7}[E^3] + k_{-8}[E_3] - [E]((k_1+k_2)[0] \\
& + (k_3+k_4)[1] + (k_5+k_6)[2] + (k_7+k_8)[3]),
\end{aligned} \tag{B.1}
$$

$$\begin{aligned}\frac{d[0]}{dt} = {} & k_{-1}([E^0]+[E^{01}]+[E^{02}]+[E^{03}]+[E_1^0]+[E_2^0]+[E_3^0]+[E_1^{01}] \\ & +[E_2^{01}]+[E_3^{01}]+[E_1^{02}]+[E_3^{02}]+[E_1^{03}]+[E_2^{03}]+[E_3^{03}]) \\ & +k_{-2}([E_0]+[E_{01}]+[E_{02}]+[E_{03}]+[E_0^1]+[E_0^2]+[E_0^3]+[E_{01}^1] \\ & +[E_{01}^2]+[E_{01}^3]+[E_{02}^1]+[E_{02}^3]+[E_{03}^1]+[E_{03}^2]+[E_{03}^3]) \\ & +(k_{-1}+k_{-2})([E_0^0]+[E_0^{01}]+[E_0^{02}]+[E_0^{03}]+[E_{01}^0]+[E_{02}^0]+[E_{03}^0] \\ & +[E_{01}^{01}]+[E_{02}^{01}]+[E_{03}^{01}]+[E_{01}^{02}]+[E_{03}^{02}]+[E_{01}^{03}]+[E_{02}^{03}]+[E_{03}^{03}]) \\ & -[0]((k_1+k_2)([E]+[E^1]+[E_1]+[E^2]+[E_2]+[E^3]+[E_3] \\ & +[E_1^1]+[E_2^1]+[E_3^1]+[E_1^2]+[E_3^2]+[E_1^3]+[E_2^3]+[E_3^3]) \\ & +k_1([E_0]+[E_0^1]+[E_0^2]+[E_0^3]+[E_{01}]+[E_{02}]+[E_{03}]+[E_{01}^1] \\ & +[E_{01}^2]+[E_{01}^3]+[E_{02}^1]+[E_{02}^3]+[E_{03}^1]+[E_{03}^2]+[E_{03}^3]) \\ & +k_2([E^0]+[E^{01}]+[E^{02}]+[E^{03}]+[E_1^0]+[E_2^0]+[E_3^0]+[E_1^{01}] \\ & +[E_2^{01}]+[E_3^{01}]+[E_1^{02}]+[E_3^{02}]+[E_1^{03}]+[E_2^{03}]+[E_3^{03}])),\end{aligned} \tag{B.2}$$

$$\begin{aligned}\frac{d[1]}{dt} = {} & k_{-3}([E^1]+[E_0^1]+[E_2^1]+[E_3^1]+[E_0^{01}]+[E_2^{01}]+[E_3^{01}]+[E_{02}^1]+[E_{03}^1] \\ & +[E_{02}^{01}]+[E_{03}^{01}])+k_{-4}([E_1]+[E_1^0]+[E_1^2]+[E_1^3]+[E_{01}^0]+[E_{01}^2] \\ & +[E_{01}^3]+[E_1^{02}]+[E_1^{03}]+[E_{01}^{02}]+[E_{01}^{03}])+(k_{-3}+k_{-4})([E_1^1]+[E_1^{01}] \\ & +[E_{01}^1]+[E_{01}^{01}])-[1]((k_3+K_4)([E]+[E^0]+[E_0]+[E_0^0])+k_3([E_1] \\ & +[E_2]+[E_3]+[E_1^0]+[E_2^0]+[E_3^0]+[E_{01}]+[E_{02}]+[E_{03}]+[E_{01}^0] \\ & +[E_{02}^0]+[E_{03}^0])+k_4([E^1]+[E^3]+[E_0^1]+[E_0^3]+[E^{01}]+[E^{03}] \\ & +[E_0^{01}]+[E_0^{03}])),\end{aligned} \tag{B.3}$$

$$\begin{aligned}\frac{d[2]}{dt} = {} & k_0([E_{01}]+[E_{01}^0]+[E_{01}^1]+[E_{01}^2]+[E_{01}^3]+[E_{01}^{01}]+[E_{01}^{02}]+[E_{01}^{03}]) \\ & +k_{-5}([E^2]+[E^{02}]+[E_0^2]+[E_1^2]+[E_3^2]+[E_0^{02}]+[E_1^{02}]+[E_3^{02}] \\ & +[E_{01}^2]+[E_{03}^2]+[E_{01}^{02}]+[E_{03}^{02}])+k_{-6}([E_2]+[E_{02}]+[E_2^0]+[E_2^1] \\ & +[E_{02}^0]+[E_{02}^1]+[E_2^{01}]+[E_{02}^{01}])+k_{-9}([E_2^3]+[E_2^{03}]+[E_{02}^3]+[E_{02}^{03}]) \\ & -[2]((k_5+k_6)([E]+[E^0]+[E_0]+[E_0^0])+k_5([E_1]+[E_1^0]+[E_3] \\ & +[E_3^0]+[E_{01}]+[E_{03}]+[E_{01}^0]+[E_{03}^0])+k_6([E^1]+[E_0^1]+[E^{01}] \\ & +[E_0^{01}])+k_9([E^3]+[E_0^3]+[E^{03}]+[E_0^{03}])),\end{aligned} \tag{B.4}$$

$$\begin{aligned}\frac{d[3]}{dt} = {} & k_{-7}([E^3]+[E^{03}]+[E_0^3]+[E_1^3]+[E_2^3]+[E_0^{03}]+[E_1^{03}]+[E_2^{03}]+[E_{01}^3] \\ & +[E_{02}^3]+[E_{01}^{03}]+[E_{02}^{03}])+k_{-8}([E_3]+[E_{03}]+[E_3^0]+[E_3^1]+[E_3^2] \\ & +[E_{03}^0]+[E_{03}^1]+[E_{03}^2]+[E_3^{01}]+[E_3^{02}]+[E_{03}^{01}]+[E_{03}^{02}]) \\ & +(k_{-7}+k_{-8})([E_3^3]+[E_3^{03}]+[E_{03}^3]+[E_{03}^{03}]) \\ & -[3](k_7([E_1]+[E_2]+[E_3]+[E_1^0]+[E_2^0]+[E_3^0]+[E_{01}]+[E_{02}]+[E_{03}] \\ & +[E_{01}^0]+[E_{02}^0]+[E_{03}^0])+k_8([E^1]+[E^3]+[E_0^1]+[E_0^3]+[E^{01}] \\ & +[E^{03}]+[E_0^{01}]+[E_0^{03}])+(k_7+k_8)([E]+[E^0]+[E_0]+[E_0^0])),\end{aligned} \tag{B.5}$$

$$\frac{d[4]}{dt} = k_0([E_{01}]+[E_{01}^0]+[E_{01}^1]+[E_{01}^2]+[E_{01}^3]+[E_{01}^{01}]+[E_{01}^{02}]+[E_{01}^{03}]), \tag{B.6}$$

$$\begin{aligned}\frac{d[E^0]}{dt} &= k_0[E^0_{01}] + k_1[0][E] + k_{-2}[E^0_0] + k_{-3}[E^{01}] + k_{-4}[E^0_1] + k_{-5}[E^{02}] \\ &\quad + k_{-6}[E^0_2] + k_{-7}[E^{03}] + k_{-8}[E^0_3] - [E^0](k_{-1} + k_2[0] \\ &\quad + (k_3 + k_4)[1] + (k_5 + k_6)[2] + (k_7 + k_8)[3]),\end{aligned}\tag{B.7}$$

$$\begin{aligned}\frac{d[E^1]}{dt} &= k_0[E^1_{01}] + k_{-1}[E^{01}] + k_{-2}[E^1_0] + k_{-4}[E^1_1] + k_{-6}[E^1_2] + k_{-8}[E^1_3] \\ &\quad + k_3[1][E] - [E^1](k_{-3} + (k_1 + k_2)[0] + k_4[1] + k_6[2] + k_8[3]),\end{aligned}\tag{B.8}$$

$$\begin{aligned}\frac{d[E^2]}{dt} &= k_0[E^2_{01}] + k_{-1}[E^{02}] + k_{-2}[E^2_0] + k_{-4}[E^2_1] + k_{-8}[E^2_3] + k_5[2][E] \\ &\quad - [E^2](k_{-5} + (k_1 + k_2)[0]),\end{aligned}\tag{B.9}$$

$$\begin{aligned}\frac{d[E^3]}{dt} &= k_0[E^3_{01}] + k_{-1}[E^{03}] + k_{-2}[E^3_0] + k_{-4}[E^3_1] + k_{-9}[E^3_2] + k_{-8}[E^3_3] \\ &\quad + k_7[3][E] - [E^3](k_{-7} + (k_1 + k_2)[0] + k_4[1] + k_9[2] + k_8[3]),\end{aligned}\tag{B.10}$$

$$\begin{aligned}\frac{d[E_0]}{dt} &= k_2[E][0] + k_{-1}[E^0_0] + k_{-3}[E^1_0] + k_{-4}[E_{01}] \\ &\quad + k_{-5}[E^2_0] + k_{-6}[E_{02}] + k_{-7}[E^3_0] + k_{-8}[E_{03}] \\ &\quad - [E_0](k_{-2} + k_1[0] + (k_3 + k_4)[1] + (k_5 + k_6)[2] + (k_7 + k_8)[3]),\end{aligned}\tag{B.11}$$

$$\begin{aligned}\frac{d[E_1]}{dt} &= k_{-1}[E^0_1] + k_{-2}[E_{01}] + k_{-3}[E^1_1] + k_{-5}[E^2_1] + k_{-7}[E^3_1] + k_4[1][E] \\ &\quad - [E_1](k_{-4} + (k_1 + k_2)[0] + k_3[1] + k_5[2] + k_7[3]),\end{aligned}\tag{B.12}$$

$$\begin{aligned}\frac{d[E_2]}{dt} &= k_{-1}[E^0_2] + k_{-2}[E_{02}] + k_{-3}[E^1_2] + k_{-7}[E^3_2] + k_6[2][E] \\ &\quad - [E_2](k_{-6} + (k_1 + k_2)[0] + k_3[1] + k_7[3]),\end{aligned}\tag{B.13}$$

$$\begin{aligned}\frac{d[E_3]}{dt} &= k_{-1}[E^0_3] + k_{-2}[E_{03}] + k_{-3}[E^1_3] + k_{-5}[E^2_3] + k_{-7}[E^3_3] + k_8[3][E] \\ &\quad - [E_3](k_{-8} + (k_1 + k_2)[0] + k_3[1] + k_5[2] + k_7[3]),\end{aligned}\tag{B.14}$$

$$\begin{aligned}\frac{d[E^{01}]}{dt} &= k_0[E^{01}_{01}] + k_{-2}[E^{01}_0] + k_{-4}[E^{01}_1] + k_{-6}[E^{01}_2] + k_{-8}[E^{01}_3] + k_1[0][E^1] \\ &\quad + k_3[1][E^0] - [E^{01}](k_{-1} + k_{-3} + k_2[0] + k_4[1] + k_6[2] + k_8[3]),\end{aligned}\tag{B.15}$$

$$\begin{aligned}\frac{d[E^{02}]}{dt} &= k_0[E^{02}_{01}] + k_{-2}[E^{02}_0] + k_{-4}[E^{02}_1] + k_{-8}[E^{02}_3] + k_1[0][E^2] + k_5[2][E^0] \\ &\quad - [E^{02}](k_{-1} + k_{-5} + k_2[0]),\end{aligned}\tag{B.16}$$

$$\begin{aligned}\frac{d[E^{03}]}{dt} &= k_0[E^{03}_{01}] + k_{-2}[E^{03}_0] + k_{-4}[E^{03}_1] + k_{-9}[E^{03}_2] + k_{-8}[E^{03}_3] + k_1[0][E^3] \\ &\quad + k_7[3][E^0] - [E^{03}](k_{-1} + k_{-7} + k_2[0] + k_4[1] + k_9[2] + k_8[3]),\end{aligned}\tag{B.17}$$

$$\begin{aligned}\frac{d[E^0_0]}{dt} &= k_{-3}[E^{01}_0] + k_{-4}[E^0_{01}] + k_{-5}[E^{02}_0] + k_{-6}[E^0_{02}] + k_{-7}[E^{03}_0] + k_{-8}[E^0_{03}] \\ &\quad + (k_1[E_0] + k_2[E^0])[0] - [E^0_0](k_{-1} + k_{-2} + (k_3 + k_4)[1] \\ &\quad + (k_5 + k_6)[2] + (k_7 + k_8)[3]),\end{aligned}\tag{B.18}$$

$$\begin{aligned}\frac{d[E_1^0]}{dt} &= k_{-2}[E_{01}^0] + k_{-3}[E_1^{01}] + k_{-5}[E_1^{02}] + k_{-7}[E_1^{03}] + k_1[0][E_1] + k_4[1][E^0] \\ &\quad - [E_1^0](k_{-1} + k_{-4} + k_2[0] + k_3[1] + k_5[2] + k_7[3]),\end{aligned} \tag{B.19}$$

$$\begin{aligned}\frac{d[E_2^0]}{dt} &= k_{-2}[E_{02}^0] + k_{-3}[E_2^{01}] + k_{-7}[E_2^{03}] + k_1[0][E_2] + k_6[2][E^0] \\ &\quad - [E_2^0](k_{-1} + k_{-6} + k_2[0] + k_3[1] + k_7[3]),\end{aligned} \tag{B.20}$$

$$\begin{aligned}\frac{d[E_3^0]}{dt} &= k_{-2}[E_{03}^0] + k_{-3}[E_3^{01}] + k_{-5}[E_3^{02}] + k_{-7}[E_3^{03}] + k_1[0][E_3] + k_8[3][E^0] \\ &\quad - [E_3^0](k_{-1} + k_{-8} + k_2[0] + k_3[1] + k_5[2] + k_7[3]),\end{aligned} \tag{B.21}$$

$$\begin{aligned}\frac{d[E_0^1]}{dt} &= k_{-1}[E_0^{01}] + k_{-4}[E_{01}^1] + k_{-6}[E_{02}^1] + k_{-8}[E_{03}^1] + k_2[0][E^1] + k_3[1][E_0] \\ &\quad - [E_0^1](k_{-2} + k_{-3} + k_1[0] + k_4[1] + k_6[2] + k_8[3]),\end{aligned} \tag{B.22}$$

$$\begin{aligned}\frac{d[E_1^1]}{dt} &= k_{-1}[E_1^{01}] + k_{-2}[E_{01}^1] + [1](k_3[E_1] + k_4[E^1]) - [E_1^1](k_{-3} + k_{-4} \\ &\quad + (k_1 + k_2)[0]),\end{aligned} \tag{B.23}$$

$$\begin{aligned}\frac{d[E_2^1]}{dt} &= k_{-1}[E_2^{01}] + k_{-2}[E_{02}^1] + k_3[1][E_2] + k_6[2][E^1] - [E_2^1](k_{-3} + k_{-6} \\ &\quad + (k_1 + k_2)[0]),\end{aligned} \tag{B.24}$$

$$\begin{aligned}\frac{d[E_3^1]}{dt} &= k_{-1}[E_3^{01}] + k_{-2}[E_{03}^1] + k_3[1][E_3] + k_8[3][E^1] - [E_3^1](k_{-3} + k_{-8} \\ &\quad + (k_1 + k_2)[0]),\end{aligned} \tag{B.25}$$

$$\begin{aligned}\frac{d[E_0^2]}{dt} &= k_{-1}[E_0^{02}] + k_{-4}[E_{01}^2] + k_{-8}[E_{03}^2] + k_2[0][E^2] + k_5[2][E_0] \\ &\quad - [E_0^2](k_{-2} + k_{-5} + k_1[0]),\end{aligned} \tag{B.26}$$

$$\frac{d[E_1^2]}{dt} = k_{-1}[E_1^{02}] + k_{-2}[E_{01}^2] + k_5[2][E_1] - [E_1^2](k_{-4} + k_{-5} + (k_1 + k_2)[0]), \tag{B.27}$$

$$\frac{d[E_3^2]}{dt} = k_{-1}[E_3^{02}] + k_{-2}[E_{03}^2] + k_5[2][E_3] - [E_3^2](k_{-5} + k_{-8} + (k_1 + k_2)[0]), \tag{B.28}$$

$$\begin{aligned}\frac{d[E_0^3]}{dt} &= k_{-1}[E_0^{03}] + k_{-4}[E_{01}^3] + k_{-9}[E_{02}^3] + k_{-8}[E_{03}^3] + k_2[0][E^3] + k_7[3][E_0] \\ &\quad - [E_0^3](k_{-2} + k_{-7} + k_1[0] + k_4[1] + k_9[2] + k_8[3]),\end{aligned} \tag{B.29}$$

$$\begin{aligned}\frac{d[E_1^3]}{dt} &= k_{-1}[E_1^{03}] + k_{-2}[E_{01}^3] + k_4[1][E^3] + k_7[3][E_1] - [E_1^3](k_{-4} + k_{-7} \\ &\quad + (k_1 + k_2)[0]),\end{aligned} \tag{B.30}$$

$$\begin{aligned}\frac{d[E_2^3]}{dt} &= k_{-1}[E_2^{03}] + k_{-2}[E_{02}^3] + k_9[2][E^3] + k_7[3][E_2] - [E_2^3](k_{-9} + k_{-7} \\ &\quad + (k_1 + k_2)[0]),\end{aligned} \tag{B.31}$$

$$\begin{aligned}\frac{d[E_3^3]}{dt} &= k_{-1}[E_3^{03}] + k_{-2}[E_{03}^3] + (k_7[E_3] + k_8[E^3])[3] - [E_3^3](k_{-7} + k_{-8} \\ &\quad + (k_1 + k_2)[0]),\end{aligned} \tag{B.32}$$

$$\begin{aligned}\frac{d[E_{01}]}{dt} &= k_{-1}[E^{0}_{01}] + k_{-3}[E^{1}_{01}] + k_{-5}[E^{2}_{01}] + k_{-7}[E^{3}_{01}] + k_{2}[0][E_{1}] + k_{4}[1][E_{0}] \\ &\quad - [E_{01}](k_{0} + k_{-2} + k_{-4} + k_{1}[0] + k_{3}[1] + k_{5}[2] + k_{7}[3]),\end{aligned} \tag{B.33}$$

$$\begin{aligned}\frac{d[E_{02}]}{dt} &= k_{-1}[E^{0}_{02}] + k_{-3}[E^{1}_{02}] + k_{-7}[E^{3}_{02}] + k_{2}[0][E_{2}] + k_{6}[2][E_{0}] \\ &\quad - [E_{02}](k_{-2} + k_{-6} + k_{1}[0] + k_{3}[1] + k_{7}[3]),\end{aligned} \tag{B.34}$$

$$\begin{aligned}\frac{d[E_{03}]}{dt} &= k_{-1}[E^{0}_{03}] + k_{-3}[E^{1}_{03}] + k_{-5}[E^{2}_{03}] + k_{-7}[E^{3}_{03}] + k_{2}[0][E_{3}] + k_{8}[3][E_{0}] \\ &\quad - [E_{03}](k_{-2} + k_{-8} + k_{1}[0] + k_{3}[1] + k_{5}[2] + k_{7}[3]),\end{aligned} \tag{B.35}$$

$$\begin{aligned}\frac{d[E^{01}_{0}]}{dt} &= k_{-4}[E^{01}_{01}] + k_{-6}[E^{01}_{02}] + k_{-8}[E^{01}_{03}] + (k_{1}[E^{1}_{0}] + k_{2}[E^{01}])[0] + k_{3}[1][E^{0}_{0}] \\ &\quad - [E^{01}_{0}](k_{-1} + k_{-2} + k_{-3} + k_{4}[1] + k_{6}[2] + k_{8}[3]),\end{aligned} \tag{B.36}$$

$$\begin{aligned}\frac{d[E^{01}_{1}]}{dt} &= k_{-2}[E^{01}_{01}] + k_{1}[0][E^{1}_{1}] + (k_{3}[E^{0}_{1}] + k_{4}[E^{01}])[1] \\ &\quad - [E^{01}_{1}](k_{-1} + k_{-3} + k_{-4} + k_{2}[0]),\end{aligned} \tag{B.37}$$

$$\begin{aligned}\frac{d[E^{01}_{2}]}{dt} &= k_{-2}[E^{01}_{02}] + k_{1}[0][E^{1}_{2}] + k_{3}[1][E^{0}_{2}] + k_{6}[2][E^{01}] \\ &\quad - [E^{01}_{2}](k_{-1} + k_{-3} + k_{-6} + k_{2}[0]),\end{aligned} \tag{B.38}$$

$$\begin{aligned}\frac{d[E^{01}_{3}]}{dt} &= k_{-2}[E^{01}_{03}] + k_{1}[0][E^{1}_{3}] + k_{3}[1][E^{0}_{3}] + k_{8}[3][E^{01}] \\ &\quad - [E^{01}_{3}](k_{-1} + k_{-3} + k_{-8} + k_{2}[0]),\end{aligned} \tag{B.39}$$

$$\begin{aligned}\frac{d[E^{02}_{0}]}{dt} &= k_{-4}[E^{02}_{01}] + k_{-8}[E^{02}_{03}] + (k_{1}[E^{2}_{0}] + k_{2}[E^{02}])[0] + k_{5}[E^{0}_{0}][2] \\ &\quad - [E^{02}_{0}](k_{-1} + k_{-2} + k_{-5}),\end{aligned} \tag{B.40}$$

$$\begin{aligned}\frac{d[E^{02}_{1}]}{dt} &= k_{-2}[E^{02}_{01}] + k_{1}[0][E^{2}_{1}] + k_{5}[2][E^{0}_{1}] \\ &\quad - [E^{02}_{1}](k_{-1} + k_{-4} + k_{-5} + k_{2}[0]),\end{aligned} \tag{B.41}$$

$$\begin{aligned}\frac{d[E^{02}_{3}]}{dt} &= k_{-2}[E^{02}_{03}] + k_{1}[0][E^{2}_{3}] + k_{5}[2][E^{0}_{3}] \\ &\quad - [E^{02}_{3}](k_{-1} + k_{-5} + k_{-8} + k_{2}[0]),\end{aligned} \tag{B.42}$$

$$\begin{aligned}\frac{d[E^{03}_{0}]}{dt} &= k_{-4}[E^{03}_{01}] + k_{-9}[E^{03}_{02}] + k_{-8}[E^{03}_{03}] + (k_{1}[E^{3}_{0}] + k_{2}[E^{03}])[0] + k_{7}[3][E^{0}_{0}] \\ &\quad - [E^{03}_{0}](k_{-1} + k_{-2} + k_{-7} + k_{4}[1] + k_{9}[2] + k_{8}[3]),\end{aligned} \tag{B.43}$$

$$\begin{aligned}\frac{d[E^{03}_{1}]}{dt} &= k_{-2}[E^{03}_{01}] + k_{1}[0][E^{3}_{1}] + k_{4}[1][E^{03}] + k_{7}[3][E^{0}_{1}] \\ &\quad - [E^{03}_{1}](k_{-1} + k_{-4} + k_{-7} + k_{2}[0]),\end{aligned} \tag{B.44}$$

$$\begin{aligned}\frac{d[E^{03}_{2}]}{dt} &= k_{-2}[E^{03}_{02}] + k_{1}[0][E^{3}_{2}] + k_{9}[2][E^{03}] + k_{7}[3][E^{0}_{2}] \\ &\quad - [E^{03}_{2}](k_{-1} + k_{-9} + k_{-7} + k_{2}[0]),\end{aligned} \tag{B.45}$$

$$\begin{aligned}\frac{d[E_3^{03}]}{dt} &= k_{-2}[E_{03}^{03}] + k_1[0][E_3^3] + (k_7[E_3^0] + k_8[E^{03}])[3] \\ &\quad - [E_3^{03}](k_{-1} + k_{-7} + k_{-8} + k_2[0]),\end{aligned} \tag{B.46}$$

$$\begin{aligned}\frac{d[E_{01}^{0}]}{dt} &= k_{-3}[E_{01}^{01}] + k_{-5}[E_{01}^{02}] + k_{-7}[E_{01}^{03}] + (k_1[E_{01}] + k_2[E_1^0])[0] + k_4[1][E_0^0] \\ &\quad - [E_{01}^0](k_0 + k_{-1} + k_{-2} + k_{-4} + k_3[1] + k_5[2] + k_7[3]),\end{aligned} \tag{B.47}$$

$$\begin{aligned}\frac{d[E_{01}^{1}]}{dt} &= k_{-1}[E_{01}^{01}] + k_2[0][E_1^1] + (k_3[E_{01}] + k_4[E_0^1])[1] \\ &\quad - [E_{01}^1](k_0 + k_{-2} + k_{-3} + k_{-4} + k_1[0]),\end{aligned} \tag{B.48}$$

$$\begin{aligned}\frac{d[E_{01}^{2}]}{dt} &= k_{-1}[E_{01}^{02}] + k_2[0][E_1^2] + k_5[2][E_{01}] \\ &\quad - [E_{01}^2](k_0 + k_{-2} + k_{-4} + k_{-5} + k_1[0]),\end{aligned} \tag{B.49}$$

$$\begin{aligned}\frac{d[E_{01}^{3}]}{dt} &= k_{-1}[E_{01}^{03}] + k_2[0][E_1^3] + k_4[1][E_0^3] + k_7[3][E_{01}] \\ &\quad - [E_{01}^3](k_0 + k_{-2} + k_{-4} + k_{-7} + k_1[0]),\end{aligned} \tag{B.50}$$

$$\begin{aligned}\frac{d[E_{02}^{0}]}{dt} &= k_{-3}[E_{02}^{01}] + k_{-7}[E_{02}^{03}] + (k_1[E_{02}] + k_2[E_2^0])[0] + k_6[2][E_0^0] \\ &\quad - [E_{02}^0](k_{-1} + k_{-2} + k_{-6} + k_3[1] + k_7[3]),\end{aligned} \tag{B.51}$$

$$\begin{aligned}\frac{d[E_{02}^{1}]}{dt} &= k_{-1}[E_{02}^{01}] + k_2[0][E_2^1] + k_3[1][E_{02}] + k_6[2][E_0^1] \\ &\quad - [E_{02}^1](k_{-2} + k_{-3} + k_{-6} + k_1[0]),\end{aligned} \tag{B.52}$$

$$\begin{aligned}\frac{d[E_{02}^{3}]}{dt} &= k_{-1}[E_{02}^{03}] + k_2[0][E_2^3] + k_9[2][E_0^3] + k_7[3][E_2^0] \\ &\quad - [E_{02}^3(k_{-2} + k_{-9} + k_{-7} + k_1[0]),\end{aligned} \tag{B.53}$$

$$\begin{aligned}\frac{d[E_{03}^{0}]}{dt} &= k_{-3}[E_{03}^{01}] + k_{-5}[E_{03}^{02}] + k_{-7}[E_{03}^{03}] + (k_1[E_{03}] + k_2[E_3^0])[0] + k_8[3][E_0^0] \\ &\quad - [E_{03}^0](k_{-1} + k_{-2} + k_{-8} + k_3[1] + k_5[2] + k_7[3]),\end{aligned} \tag{B.54}$$

$$\begin{aligned}\frac{d[E_{03}^{1}]}{dt} &= k_{-1}[E_{03}^{01}] + k_2[0][E_3^1] + k_3[1][E_{03}] + k_8[3][E_0^1] \\ &\quad - [E_{03}^1](k_{-2} + k_{-3} + k_{-8} + k_1[0]),\end{aligned} \tag{B.55}$$

$$\begin{aligned}\frac{d[E_{03}^{2}]}{dt} &= k_{-1}[E_{03}^{02}] + k_2[0][E_3^2] + k_5[2][E_{03}] \\ &\quad - [E_{03}^2](k_{-2} + k_{-5} + k_{-8} + k_1[0]),\end{aligned} \tag{B.56}$$

$$\begin{aligned}\frac{d[E_{03}^{3}]}{dt} &= k_{-1}[E_{03}^{03}] + k_2[0][E_3^3] + (k_7[E_{03}] + k_8[E_0^3])[3] \\ &\quad - [E_{03}^3](k_{-2} + k_{-7} + k_{-8} + k_1[0]),\end{aligned} \tag{B.57}$$

$$\begin{aligned}\frac{d[E_{01}^{01}]}{dt} &= (k_1[E_{01}^1] + k_2[E_1^{01}])[0] + (k_3[E_{01}^0] + k_4[E_0^{01}])[1] \\ &\quad - [E_{01}^{01}](k_0 + k_{-1} + k_{-2} + k_{-3} + k_{-4}),\end{aligned} \tag{B.58}$$

$$\begin{aligned}
\frac{d[E^{01}_{02}]}{dt} &= (k_1[E^{1}_{02}] + k_2[E^{01}_{2}])[0] + k_3[1][E^{0}_{02}] + k_6[2][E^{01}_{0}] \\
&\quad - [E^{01}_{02}](k_{-1} + k_{-2} + k_{-3} + k_{-6}), \qquad \text{(B.59)} \\
\frac{d[E^{01}_{03}]}{dt} &= (k_1[E^{1}_{03}] + k_2[E^{01}_{3}])[0] + k_3[1][E^{0}_{03}] + k_8[3][E^{01}_{0}] \\
&\quad - [E^{01}_{03}](k_{-1} + k_{-2} + k_{-3} + k_{-8}), \qquad \text{(B.60)} \\
\frac{d[E^{02}_{01}]}{dt} &= (k_1[E^{2}_{01}] + k_2[E^{02}_{1}])[0] + k_5[2][E^{0}_{01}] \\
&\quad - [E^{02}_{01}](k_0 + k_{-1} + k_{-2} + k_{-4} + k_{-5}), \qquad \text{(B.61)} \\
\frac{[E^{02}_{03}]}{dt} &= (k_1[E^{2}_{03}] + k_2[E^{02}_{3}])[0] + k_5[2][E^{0}_{03}] \\
&\quad - [E^{02}_{03}](k_{-1} + k_{-2} + k_{-5} + k_{-8}), \qquad \text{(B.62)} \\
\frac{[E^{03}_{01}]}{dt} &= (k_1[E^{3}_{01}] + k_2[E^{03}_{1}])[0] + k_4[1][E^{03}_{0}] + k_7[3][E^{0}_{01}] \\
&\quad - [E^{03}_{01}](k_0 + k_{-1} + k_{-2} + k_{-4} + k_{-7}), \qquad \text{(B.63)} \\
\frac{d[E^{03}_{02}]}{dt} &= (k_1[E^{3}_{02}] + k_2[E^{03}_{2}])[0] + k_9[2][E^{03}_{0}] + k_7[3][E^{0}_{02}] \\
&\quad - [E^{03}_{02}](k_{-1} + k_{-2} + k_{-9} + k_{-7}), \qquad \text{(B.64)} \\
\frac{d[E^{03}_{03}]}{dt} &= (k_1[E^{3}_{03}] + k_2[E^{03}_{3}])[0] + (k_7[E^{0}_{03}] + k_8[E^{03}_{0}])[3] \\
&\quad - [E^{03}_{03}](k_{-1} + k_{-2} + k_{-7} + k_{-8}). \qquad \text{(B.65)}
\end{aligned}$$

These equations are solved subject to the initial conditions

$$\begin{aligned}
&[E](t=0) = E_0, \\
&[0](t=0) = G_0, \\
&[1](t=0) = ATP_0, \\
&[2](t=0) = 0, \\
&[3](t=0) = P_{i0}, \\
&[4](t=0) = 0, \\
&[E^k](t=0) = 0, && k = 0,1,2,3, \\
&[E_k](t=0) = 0, && k = 0,1,2,3, \\
&[E^{0j}](t=0) = 0, && j = 1,2,3, \\
&[E_{0j}](t=0) = 0, && j = 1,2,3, \\
&[E^{k}_{j}](t=0) = 0, && k = 0,1,2,3, && j = 0,1,3, \\
&[E^{j}_{2}](t=0) = 0, && j = 0,1,3, \\
&[E^{0x}_{y}](t=0) = 0, && x = 1,2,3, && y = 0,1,3, \\
&[E^{x}_{0y}](t=0) = 0, && x = 0,1,3, && y = 1,2,3,
\end{aligned}$$

$$
\begin{aligned}
&[E_2^{0x}](t=0)=0, && x=1,3, &&\\
&[E_{0x}^{2}](t=0)=0, && x=1,3, &&\\
&[E_{0y}^{0x}](t=0)=0, && x=1,3, && y=1,3,\\
&[E_{0x}^{02}](t=0)=0, && x=1,3, &&\\
&[E_{02}^{0x}](t=0)=0, && x=1,3, &&
\end{aligned}
$$

where E_0, G_0, ATP_0, P_{i0} give the initial concentrations of enzyme, glucose, ATP, and P_i, respectively.

B.3 Software

We display three files: the file which implements the Sensitivity Analysis, the main file which calculates and plots solutions by calling the model file containing the model equations. See the NEXT PAGE.

```
#======================#
# SENSITIVITY ANALYSIS #
#======================#

from SALib.sample import saltelli
from SALib.analyze import sobol
import numpy as np
from scipy.integrate import odeint
from scipy import integrate

# SOLUTION TO THE MODEL
#
def SOL(p, initial_cond, t0, t_end, stpz):
    #
    k0, k1, k_1, k2, k_2, k3, k_3, k4, k_4, k5, k_5, \
    k6, k_6, k7, k_7, k8, k_8, k9, k_9 = p
    #
    t = np.arange(t0, t_end, stpz)
    #
    def MODEL(y,t):
        #
        #
        # Define y

        _E_, _0_, _1_, _2_, _3_, _4_,\
            \
        _0E_,_1E_, _2E_, _3E_,\
            \
        _E0_, _E1_, _E2_, _E3_,\
            \
        _01E_, _02E_, _03E_,\
            \
        _0E0_, _0E1_, _0E2_, _0E3_,\
            \
        _1E0_, _1E1_, _1E2_, _1E3_,\
            \
        _2E0_, _2E1_, _2E3_,\
            \
        _3E0_, _3E1_, _3E2_, _3E3_,\
            \
        _E01_, _E02_, _E03_,\
            \
        _01E0_, _01E1_, _01E2_, _01E3_,\
            \
        _02E0_, _02E1_, _02E3_,\
            \
        _03E0_, _03E1_, _03E2_, _03E3_,\
            \
        _0E01_, _1E01_, _2E01_, _3E01_,\
            \
        _0E02_, _1E02_, _3E02_,\
            \
        _0E03_, _1E03_, _2E03_, _3E03_,\
            \
        _01E01_, _01E02_, _01E03_, _02E01_,\
            \
        _02E03_, _03E01_, _03E02_, _03E03_ = y
        #
        #
        # _E_: Hexokinase 1; _0_: Glucose
        # _1_: ATP; _2_: G6P; _3_: Pi; _4_: ADP
        # _xEy_: x, y substances bound at N, C domains, respectively.
        # Define dydt
        dydt=[]
        #
        #1 Eq. for enyme _E_
        dydt.append(k0*_E01_ + k_1*_0E_ + k_2*_E0_ + k_3*_1E_ \

        + k_4*_E1_ + k_5*_2E_ + k_6*_E2_ + k_7*_3E_ + k_8*_E3_ \
        - _E_*((k1 + k2)*_0_ + (k3 + k4)*_1_ + (k5 + k6)*_2_ \
            + (k7 + k8)*_3_))
        #2 Eq. for _0_
        dydt.append(k_1*(_0E_ + _01E_ + _02E_ + _03E_ + _0E0_ \
            + _0E1_ + _0E2_ + _0E3_ + _01E0_ + _01E1_ + _01E2_ \
            + _01E3_ + _02E0_ + _02E1_ + _02E3_ + _03E0_ \
            + _03E1_ + _03E2_ + _03E3_ + _0E01_ + _0E02_ \
            + _0E03_ + _01E01_ + _01E02_ + _01E03_ + _02E01_ \
            + _02E03_ + _03E01_ + _03E02_ + _03E03_ )\
        + k_2*(_E0_ + _0E0_ + _1E0_ + _2E0_ + _3E0_ + _E01_ \
            + _E02_ + _E03_ + _01E0_ + _02E0_ + _03E0_ + _0E01_ \
            + _1E01_ + _2E01_ + _3E01_ +_0E02_ + _1E02_ + _3E02_ \
            + _0E03_ + _1E03_ + _2E03_ + _3E03_ + _01E01_ \
            + _01E02_ + _01E03_ + _02E01_ + _02E03_ + _03E01_ \
            + _03E02_ + _03E03_ )\
        - _0_*((k1 + k2)*(_E_ + _1E_ + _E1_ + _2E_ + _E2_ + _3E_ \
                + _E3_ + _1E1_ + _1E2_ + _1E3_ + _2E1_ + _2E3_ \
                + _3E1_ + _3E2_ + _3E3_)\
            + k1*(_E0_ + _1E0_ + _2E0_ + _3E0_ + _E01_ + _E02_ \
                + _E03_ + _1E01_ + _2E01_ + _3E01_ + _1E02_ \
                + _3E02_ + _1E03_ + _2E03_ + _3E03_) \
            + k2*(_0E_ + _01E_ + _02E_ + _03E_ + _0E1_ + _0E2_ \
                + _0E3_ + _01E1_ + _01E2_ + _01E3_ + _02E1_ \
                + _02E3_ + _03E1_ + _03E2_ + _03E3_)))
        #3 Eq. for _1_
        dydt.append(k_3*(_1E_ + _01E_ + _1E0_ + _1E1_ + _1E2_ + _1E3_ \
            + _01E0_ + _01E1_ + _01E2_ + _01E3_ + _1E01_ + _1E02_ \
            + _1E03_ + _01E01_ + _01E02_ + _01E03_) \
        + k_4*(_E1_ + _0E1_ + _1E1_ + _2E1_ + _3E1_ + _E01_ \
            + _01E1_ + _02E1_ + _03E1_ + _0E01_ + _1E01_ \
            + _2E01_ + _3E01_ + _01E01_ + _02E01_ + _03E01_) \
        - _1_*(k3*(_E_ + _0E_ + _E0_ + _E1_ + _E2_ + _E3_ + _0E0_ \
                + _0E1_ + _0E2_ + _0E3_ + _E01_ + _E02_ + _E03_ \
                + _0E01_ + _0E02_ + _0E03_) \
            + k4*(_E_ + _0E_ + _1E_ + _3E_ + _E0_ + _01E_ + _03E_ \
                + _0E0_ + _1E0_ + _3E0_ + _01E0_ + _03E0_)))

        #4 Eq. for _2_
        dydt.append(k0*(_E01_ + _0E01_ + _1E01_ + _2E01_ + _3E01_ \

                + _01E01_ + _02E01_ + _03E01_) \

            + k_5*(_2E_ + _02E_ + _2E0_ + _2E1_ + _2E3_ + _02E0_ \

                + _02E1_ + _02E3_ + _2E01_ + _2E03_ + _02E01_ \

                + _02E03_) \

            + k_6*(_E2_ + _0E2_ + _1E2_ + _E02_ + _01E2_ \

                + _0E02_ + _1E02_ + _01E02_) \

            + k_9*(_3E2_ + _03E2_ + _3E02_ +  _03E02_ ) \

        - _2_*(k5*(_E_ + _0E_ + _E0_ + _E1_ + _E3_ + _0E0_ \

                + _0E1_ + _0E3_ + _E01_ + _E03_ + _0E01_ + _0E03_) \

            + k6*(_E_ + _0E_ + _1E_ + _E0_ + _01E_ + _0E0_ \

                + _1E0_ + _01E0_) \

            + k9*(_3E_ + _03E_ + _3E0_ + _03E0_)))

        #5 Eq. for _3_
        dydt.append(k_7*(_3E_ + _03E_ + _3E0_ + _3E1_ + _3E2_ + _3E3_ \

            + _03E0_ + _03E1_ + _03E2_ + _03E3_ + _3E01_ \

            + _3E02_ + _3E03_ + _03E01_ + _03E02_ + _03E03_) \

        + k_8*(_E3_ + _0E3_ + _1E3_ + _2E3_ + _3E3_ + _E03_ \

            + _01E3_ + _02E3_ + _03E3_ + _0E03_ + _1E03_ \

            + _2E03_ + _3E03_ + _01E03_ + _02E03_ + _03E03_) \

        - _3_*(k7*(_E_ + _0E_ + _E0_ + _E1_ + _E2_ + _E3_ + _0E0_ \

                + _0E1_ + _0E2_ + _0E3_ + _E01_ + _E02_ + _E03_ \

                + _0E01_ + _0E02_ + _0E03_) \

            + k8*(_E_ + _0E_ + _1E_ + _3E_ + _E0_ + _01E_ + _03E_ \

                + _0E0_ + _1E0_ + _3E0_ + _01E0_ + _03E0_)))

        #6 Eq. for _4_
        dydt.append(k0*(_E01_ + _0E01_ + _1E01_ + _2E01_\
                + _3E01_ + _01E01_ + _02E01_ + _03E01_))
        #
        #
        #
        #7 Eq. for _0E_
        dydt.append(k0*_0E01_ + k1*_0_*_E_ + k_2*_0E0_ + k_3*_01E_ \

            + k_4*_0E1_ + k_5*_02E_ + k_6*_0E2_ + k_7*_03E_ \

            + k_8*_0E3_ \

        - _0E_*(k_1 + k2*_0_ + (k3 + k4)*_1_ + (k5 + k6)*_2_ \

            + (k7 + k8)*_3_))
```

```
#8 Eq. for _1E_
dydt.append(k0*_1E01_ + k_1*_01E_ + k_2*_1E0_ + k_4*_1E1_ \

    + k_6*_1E2_ + k_8*_1E3_ + _1_*k3*_E_ \

- _1E_*(k_3 + (k1 + k2)*_0_ + k4*_1_ + k6*_2_ + k8*_3_))

#9 Eq. for _2E_
dydt.append(k0*_2E01_ + k_1*_02E_ + k_2*_2E0_ + k_4*_2E1_ \

    + k_8*_2E3_ + _2_*k5*_E_ \

- _2E_*(k_5 + (k1 + k2)*_0_))

#10 Eq. for _3E_
dydt.append(k0*_3E01_ + k_1*_03E_ + k_2*_3E0_ + k_4*_3E1_ \

    + k_9*_3E2_ + k_8*_3E3_ + _3_*k7*_E_ \

- _3E_*(k_7 + (k1 + k2)*_0_ + k4*_1_ + k9*_2_ + k8*_3_))

#11 Eq. for _E0_
dydt.append(k2*_E_*_0_ + k_1*_0E0_ + k_3*_1E0_ + k_5*_2E0_ \

    + k_7*_3E0_  + k_4*_E01_ + k_6*_E02_  + k_8*_E03_ \

- _E0_*(k_2 + k1*_0_ + (k3 + k4)*_1_ + _2_*(k5 + k6) \

    + (k7 + k8)*_3_))

#12 Eq. for _E1_
dydt.append(k_1*_0E1_ + k_2*_E01_ + k_3*_1E1_ + k_5*_2E1_ \

    + k_7*_3E1_ + _1_*_E_*k4 \

- _E1_*(k_4 + (k1 + k2)*_0_ + k3*_1_ + k5*_2_ + k7*_3_))

#13 Eq. for _E2_
dydt.append(k_1*_0E2_ + k_2*_E02_ + k_3*_1E2_ + k_7*_3E2_ \

    + _2_*k6*_E_ \

-_E2_*(k_6 + (k1 + k2)*_0_ + k3*_1_ + k7*_3_))

#14 Eq. for _E3_
dydt.append(k_1*_0E3_ + k_2*_E03_ + k_3*_1E3_ + k_5*_2E3_ \

    + k_7*_3E3_ + _3_*k8*_E_ \

- _E3_*(k_8 + (k1 + k2)*_0_ + k3 *_1_ + k5*_2_ + k7*_3_))

#
#
#15 Eq. for _01E_
dydt.append(k0*_01E01_ + k_2*_01E0_ + k_4*_01E1_ \

    + k_6*_01E2_ + k_8*_01E3_ + _0_*_1E_*k1 + _1_*k3*_0E_ \

- _01E_*(k_1 + k_3 + k2*_0_ + k4*_1_ + k6*_2_ + k8*_3_))

#16 Eq. for _02E_
dydt.append(k0*_02E01_ + k_2*_02E0_ + k_4*_02E1_ \

    + k_8*_02E3_ + k1*_0_*_2E_ + _2_*k5*_0E_ \

- _02E_*(k_1 + k_5 + k2*_0_))

#17 Eq. for _03E_
dydt.append(k0*_03E01_ + k_2*_03E0_ + k_4*_03E1_ + k_9*_03E2_ \
    + k_8*_03E3_ + _0_*_3E_*k1 + _3_*k7*_0E_ \

- _03E_*(k_1 + k_7 + k2*_0_ + k4*_1_ + k9*_2_ + k8*_3_))

#
#18 Eq. for _0E0_
dydt.append(k_3*_01E0_ + k_4*_0E01_ + k_5*_02E0_ + k_6*_0E02_ \

    + k_7*_03E0_ + k_8*_0E03_ + _0_*(k1*_E0_ + k2*_0E_) \

- _0E0_*(k_1 + k_2 + (k3 + k4)*_1_ + (k5 + k6)*_2_ \

    + (k7 + k8)*_3_))

#19 Eq. for _0E1_
dydt.append(k_2*_0E01_ + k_3*_01E1_ + k_5*_02E1_ + k_7*_03E1_ \

    + k1*_0_*_E1_ + _1_*k4*_0E_ \

- _0E1_*(k_1 + k_4 + k2*_0_ + k3*_1_ + k5*_2_ + k7*_3_))

#20 Eq. for _0E2_
dydt.append(k_2*_0E02_ + k_3*_01E2_ + k_7*_03E2_ \

    + k1*_0_*_E2_ + _2_*k6*_0E_ \

- _0E2_*(k_1 + k_6 + k2*_0_ + k3*_1_ + k7*_3_))

#21 Eq. for _0E3_
dydt.append(k_2*_0E03_ + k_3*_01E3_ + k_5*_02E3_ \

    + k_7*_03E3_ + k1*_0_*_E3_ + _3_*k8*_0E_ \

- _0E3_*(k_1 + k_8 + k2*_0_ + k3*_1_ + k5*_2_ + k7*_3_))

#
#22 Eq. for _1E0_
dydt.append (k_1*_01E0_ + k_4*_1E01_ + k_6*_1E02_ \

    + k_8*_1E03_ + k2*_0_*_1E_ + _1_*k3*_E0_ \

- _1E0_*(k_2 + k_3 + k1*_0_ + k4*_1_ + k6*_2_ + k8*_3_))

#23 Eq. for _1E1_
dydt.append(k_1*_01E1_ + k_2*_1E01_ + _1_*(k3*_E1_ + k4*_1E_) \

    - _1E1_*(k_3 + k_4 + (k1 + k2)*_0_))

#24 Eq. for _1E2_
dydt.append(k_1*_01E2_ + k_2*_1E02_ + _1_*k3*_E2_ \

    + _2_*k6*_1E_ \

    - _1E2_*(k_3 + k_6 + (k1 + k2)*_0_))

#25 Eq. for _1E3_
dydt.append(k_1*_01E3_ + k_2*_1E03_ \

    + _1_*k3*_E3_ + _3_*k8*_1E_ \

    - _1E3_*(k_3 + k_8 + (k1 + k2)*_0_))

#
#26 Eq. for _2E0_
dydt.append(k_1*_02E0_ + k_4*_2E01_ + k_8*_2E03_ \

    + k2*_0_*_2E_ + _2_*k5*_E0_ \
    - _2E0_*(k_2 + k_5 + k1*_0_))

#27 Eq. for _2E1_
dydt.append(k_1*_02E1_ + k_2*_2E01_ + _2_*k5*_E1_ \

    - _2E1_*(k_4 + k_5 + (k1 + k2)*_0_))

#28 Eq. for _2E3_
dydt.append(k_1*_02E3_ + k_2*_2E03_ + _2_*k5*_E3_ \

    - _2E3_*(k_5 + k_8 + (k1 + k2)*_0_))

#
#29 Eq. for _3E0_
dydt.append(k_1*_03E0_ + k_4*_3E01_ + k_9*_3E02_ \

    + k_8*_3E03_ + _0_*k2*_3E_ + _3_*k7*_E0_ \

- _3E0_*(k_2 + k_7 + k1*_0_ + k4*_1_ + k9*_2_ + k8*_3_))

#30 Eq. for _3E1_
dydt.append(k_1*_03E1_ + k_2*_3E01_ + _1_*k4*_3E_ \

    + _3_*k7*_E1_ \

    - _3E1_*(k_4 + k_7 + (k1 + k2)*_0_))

#31 Eq. for _3E2_
dydt.append(k_1*_03E2_ + k_2*_3E02_ + _2_*k9*_3E_ \

    + _3_*k7*_E2_ \

    - _3E2_*(k_7 + k_9 + (k1 + k2)*_0_))

#32 Eq. for _3E3_
dydt.append(k_1*_03E3_ + k_2*_3E03_ \

    + _3_*(k7*_E3_ + k8*_3E_) \

    - _3E3_*(k_7 + k_8 + (k1 + k2)*_0_))

#
#33 Eq. for _E01_
dydt.append(k_1*_0E01_ + k_3*_1E01_ + k_5*_2E01_ \

    + k_7*_3E01_ + _0_*k2*_E1_ + _1_*k4*_E0_ \

- _E01_*(k0 + k_2 + k_4 + k1*_0_ + k3*_1_ + k5*_2_ + k7*_3_))

#34 Eq. for _E02_
dydt.append(k_1*_0E02_ + k_3*_1E02_ + k_7*_3E02_ \

    + _0_*k2*_E2_ + _2_*k6*_E0_ \

    - _E02_*(k_2 + k_6 + k1*_0_ + k3*_1_ + k7*_3_))

#35 Eq. for _E03_
dydt.append(k_1*_0E03_ + k_3*_1E03_ + k_5*_2E03_ \

    + k_7*_3E03_ + _0_*k2*_E3_ + _3_*k8*_E0_ \

- _E03_*(k_2 + k_8 + _0_*k1 + _1_*k3 + _2_*k5 + _3_*k7))

#
#
#36 Eq. for _01E0_
dydt.append(k_4*_01E01_ + k_6*_01E02_ + k_8*_01E03_ \

    + _0_*(k1*_1E0_ + k2*_01E_) + _1_*k3*_0E0_ \
- _01E0_*(k_1 + k_2 + k_3 + _1_*k4 + _2_*k6 + _3_*k8))

#37 Eq. for _01E1_
dydt.append(k_2*_01E01_ + _0_*k1*_1E1_ \

    + _1_*(k3*_0E1_ + k4*_01E_) \

    - _01E1_*(k_1 + k_3 + k_4 + _0_*k2))

#38 Eq. for _01E2_
dydt.append(k_2*_01E02_ + _0_*k1*_1E2_ + _1_*k3*_0E2_ \

    + _2_*k6*_01E_ \

    - _01E2_*(k_1 + k_3 + k_6 + _0_*k2))

#39 Eq. for _01E3_
dydt.append(k_2*_01E03_ + _0_*k1*_1E3_ \

    + _1_*k3*_0E3_ + _3_*k8*_01E_ \

    - _01E3_*(k_1 + k_3 + k_8 + _0_*k2 ))

#
#40 Eq. for _02E0_
dydt.append(k_4*_02E01_ + k_8*_02E03_ \

    + _0_*(k1*_2E0_ + k2*_02E_) + _2_*k5*_0E0_ \

- _02E0_*(k_1 + k_2 + k_5))

#41 Eq. for _02E1_
dydt.append(k_2*_02E01_ + _0_*k1*_2E1_ + _2_*k5*_0E1_ \

    - _02E1_*(k_1 + k_4 + k_5 + _0_*k2))

#42 Eq. for _02E3_
dydt.append(k_2*_02E03_ + _0_*k1*_2E3_ + _2_*k5*_0E3_ \

    - _02E3_*(k_1 + k_5 + k_8 + _0_*k2))

#
#43 Eq. for _03E0_
dydt.append(k_4*_03E01_ + k_9*_03E02_ + k_8*_03E03_ \

    + _0_*(k1*_3E0_ + k2*_03E_) + _3_*k7*_0E0_ \

- _03E0_*(k_1 + k_2 + k_7 + _1_*k4 + _2_*k9 + _3_*k8))

#44 Eq. for _03E1_
dydt.append(k_2*_03E01_ + _0_*k1*_3E1_ + _1_*k4*_03E_ \

    + _3_*k7*_0E1_ \

    - _03E1_*(k_1 + k_4 + k_7 + _0_*k2))

#45 Eq. for _03E2_
dydt.append(k_2*_03E02_ + _0_*k1*_3E2_ \

    + _2_*k9*_03E_ + _3_*k7*_0E2_ \

    - _03E2_*(k_1 + k_9 + k_7 + _0_*k2))

#46 Eq. for _03E3_
dydt.append(k_2*_03E03_ + _0_*k1*_3E3_ \

    + _3_*(k7*_0E3_ + k8*_03E_) \
    - _03E3_*(k_1 + k_7 + k_8 + _0_*k2))

#
#47 Eq. for _0E01_
dydt.append(k_3*_01E01_ + k_5*_02E01_ + k_7*_03E01_ \

    + _0_*(k1*_E01_ + k2*_0E1_) + _1_*k4*_0E0_ \

- _0E01_*(k0 + k_1 + k_2 + k_4 + _1_*k3 + _2_*k5 + _3_*k7))

#48 Eq. for _1E01_
dydt.append(k_1*_01E01_ + _0_*k2*_1E1_ \

    + _1_*(k3*_E01_ + k4*_1E0_) \

    - _1E01_*(k0 + k_2 + k_3 + k_4 + _0_*k1))

#49 Eq. for _2E01_
dydt.append(k_1*_02E01_ + _0_*k2*_2E1_ + _2_*k5*_E01_ \

    - _2E01_*(k0 + k_2 + k_4 + k_5 + _0_*k1))

#50 Eq. for _3E01_
dydt.append(k_1*_03E01_ + _0_*k2*_3E1_ + _1_*k4*_3E0_ \

    + _3_*k7*_E01_ \

    - _3E01_*(k0 + k_2 + k_4 + k_7 + _0_*k1))

#
#51 Eq. for _0E02_
dydt.append(k_3*_01E02_ + k_7*_03E02_

    + _0_*(k1*_E02_ + k2*_0E2_) + _2_*k6*_0E0_ \

    - _0E02_*(k_1 + k_2 + k_6 + _1_*k3 + _3_*k7))

#52 Eq. for _1E02_
dydt.append(k_1*_01E02_ + _0_*k2*_1E2_ + _1_*k3*_E02_ \

    + _2_*k6*_1E0_ \

    - _1E02_*(k_2 + k_3 + k_6 + _0_*k1))

#53 Eq. for _3E02_
dydt.append(k_1*_03E02_ + _0_*k2*_3E2_ + _2_*k9*_3E0_ \

    + _3_*k7*_E02_ \

    - _3E02_*(k_2 + k_9 + k_7 + _0_*k1))

#
#54 Eq. for _0E03_
dydt.append(k_3*_01E03_ + k_5*_02E03_ + k_7*_03E03_ \

    + _0_*(k1*_E03_ + k2*_0E3_) + _3_*k8*_0E0_ \

    - _0E03_*(k_1 + k_2 + k_8 + _1_*k3 + _2_*k5 + _3_*k7))

#55 Eq. for _1E03_
dydt.append(k_1*_01E03_ + _0_*k2*_1E3_ + _1_*k3*_E03_ \

    + _3_*k8*_1E0_ \

    - _1E03_*(k_2 + k_3 + k_8 + _0_*k1))

#56 Eq. for _2E03_
dydt.append(k_1*_02E03_ + _0_*k2*_2E3_ + _2_*k5*_E03_ \
    - _2E03_*(k_2 + k_5 + k_8 + _0_*k1))

#57 Eq. for _3E03_
dydt.append(k_1*_03E03_ + _0_*k2*_3E3_ \

    + _3_*(k7*_E03_ + k8*_3E0_) \

    - _3E03_*(k_2 + k_7 + k_8 + _0_*k1))

#
#
#58 Eq. for _01E01_
dydt.append(_0_*(k1*_1E01_ + k2*_01E1_) \

    + _1_*(k3*_0E01_ + k4*_01E0_) \

    - _01E01_*(k0 + k_1 + k_2 + k_3 + k_4))

#59 Eq. for _01E02_
dydt.append(_0_*(k1*_1E02_ + k2*_01E2_) \

    + _1_*k3*_0E02_ + _2_*k6*_01E0_ \

    - _01E02_*(k_1 + k_2 + k_3 + k_6))

#60 Eq. for _01E03_
dydt.append(_0_*(k1*_1E03_ + k2*_01E3_) \

    + _1_*k3*_0E03_  + _3_*k8*_01E0_ \

    - _01E03_*(k_1 + k_2 + k_3 + k_8))

#
#61 Eq. for _02E01_
dydt.append(_0_*(k1*_2E01_ + k2*_02E1_) + _2_*k5*_0E01_ \

    - _02E01_*(k0 + k_1 + k_2 + k_4 + k_5))

#62 Eq. for _02E03_
dydt.append(_0_*(k1*_2E03_ + k2*_02E3_) + _2_*k5*_0E03_ \

    - _02E03_*(k_1 + k_2 + k_5 + k_8))

#
#63 Eq. for _03E01_
dydt.append(_0_*(k1*_3E01_ + k2*_03E1_) \

    + _1_*k4*_03E0_ + _3_*k7*_0E01_ \

    - _03E01_*(k0 + k_1 + k_2 + k_4 + k_7))

#64 Eq. for _03E02_
dydt.append(_0_*(k1*_3E02_ + k2*_03E2_) \

    + _2_*k9*_03E0_ + _3_*k7*_0E02_ \

    - _03E02_*(k_1 + k_2 + k_9 + k_7))

#65 Eq. for _03E03_
dydt.append(_0_*(k1*_3E03_ + k2*_03E3_) \

    + _3_*(k7*_0E03_ + k8*_03E0_) \

    - _03E03_*(k_1 + k_2 + k_7 + k_8))

#--------------#
# Return dydt  #
#--------------#
return(dydt)
```

```
    #----------------------#
    # Integrate the model  #
    #----------------------#
    ds = integrate.odeint(MODEL, initial_cond, t)
    return(ds)
#


# -------------------#
# INITIAL CONDITIONS #
#     Unit: mM       #
#--------------------#

# E, 0, 1, 2, 3, 4
E, G, ATP, ADP = 6.65e-2, 2.5, 3.0, 0.0

G6P = 2.0
Pi = [2.0, 10.0]

y0 = [[E, G, ATP, G6P, Pi[0], ADP,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0],\
     [E, G, ATP, G6P, Pi[1], ADP,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0]]
#
#===================================#
# PARAMETER VALUES FOR SIMULATIONS  #
#===================================#

k0 = 63.0

KmG = 0.053

a = 99.0
# For Glucose
k1, k_1 = (a + 1.0)*k0/KmG, a*k0

k2, k_2 = k1, k_1

# For ATP
# Km
KmA = 0.7
k3, k_3 = (a + 1.0)*k0/KmA, a*k0

# Ki =
k4, k_4 = k3, k_3

# For G6P

# Ki = 0.71 mM (N)
k5, k_5 = k3, 0.71*k3
```

```
# Ki = 54 microM = 0.054 mM(C)
k6, k_6 = k3, 0.054*k3

#d = 1.0e-0
# For Pi
# K_i = 0.022mM
k7, k_7 = k3, 0.022*k3

# Ki = 0.22mM
k8, k_8 = k3, 0.22*k3

# For G6P binding to _3E_, _03E_, and _03E0_
k9, k_9 = 0.1*k3, 0.01*k3

#===========================================#
#  MODEL FOR SENSITIVITY ANALYSIS           #
#===========================================#
## Model for sensitivity analysis
def model(p):
    ys = SOL(p, y0[0], 0.0, 10.05, 0.01)
    #Fs = ys[:,3]
    return(ys[:,3])
    #
#====================================#
a = 0.9
b = 1.1
## Define the problem of SA
problem = {
    'num_vars': 19,
    'names':['k0', 'k1', 'k_1', 'k2', 'k_2', 'k3', 'k_3', \
        'k4', 'k_4', 'k5', 'k_5', 'k6', 'k_6', 'k7', 'k_7', \
        'k8', 'k_8', 'k9', 'k_9'],
    'bounds': [[a*k0, b*k0], \
        [a*k1, b*k1], [a*k_1, b*k_1], \
        [a*k2, b*k2], [a*k_2, b*k_2], \
        [a*k3, b*k3], [a*k_3, b*k_3], \
        [a*k4, b*k4], [a*k_4, b*k_4], \
        [a*k5, b*k5], [a*k_5, b*k_5], \
        [a*k6, b*k6], [a*k_6, b*k_6], \
        [a*k7, b*k7], [a*k_7, b*k_7], \
        [a*k8, b*k8], [a*k_8, b*k_8], \
        [a*k9, b*k9], [a*k_9, b*k_9]]
}
#
#print(problem['bounds'])
#=====================================#
## Generate samples
### Number of samples
n = 1000
##
param_values = saltelli.sample(problem, n,\
                               calc_second_order=False)

#=====================================#
#
# Data poits for the GSA
#
#L = [101, 201, 301, 401, 501, 601, 701, 801, 901, 1001]

Y1 = np.zeros([param_values.shape[0]])

Y2 = np.zeros([param_values.shape[0]])

Y3 = np.zeros([param_values.shape[0]])

Y4 = np.zeros([param_values.shape[0]])

Y5 = np.zeros([param_values.shape[0]])
```

```
Y6 = np.zeros([param_values.shape[0]])

Y7 = np.zeros([param_values.shape[0]])

Y8 = np.zeros([param_values.shape[0]])

Y9 = np.zeros([param_values.shape[0]])

Y10 = np.zeros([param_values.shape[0]])

#===========================================#
for j, X in enumerate(param_values):
    K      = model(X)
    Y1[j]  = K[101]
    Y2[j]  = K[201]
    Y3[j]  = K[301]
    Y4[j]  = K[401]
    Y5[j]  = K[501]
    Y6[j]  = K[601]
    Y7[j]  = K[701]
    Y8[j]  = K[801]
    Y9[j]  = K[901]
    Y10[j] = K[1001]

#===========================================#
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8
Y9
Y10

#===========================================#
np.savetxt(str(Pi[0]) + "Pi_Y1_outputs.txt", Y1)
np.savetxt(str(Pi[0]) + "Pi_Y2_outputs.txt", Y2)
np.savetxt(str(Pi[0]) + "Pi_Y3_outputs.txt", Y3)
np.savetxt(str(Pi[0]) + "Pi_Y4_outputs.txt", Y4)
np.savetxt(str(Pi[0]) + "Pi_Y5_outputs.txt", Y5)
np.savetxt(str(Pi[0]) + "Pi_Y6_outputs.txt", Y6)
np.savetxt(str(Pi[0]) + "Pi_Y7_outputs.txt", Y7)
np.savetxt(str(Pi[0]) + "Pi_Y8_outputs.txt", Y8)
np.savetxt(str(Pi[0]) + "Pi_Y9_outputs.txt", Y9)
np.savetxt(str(Pi[0]) + "Pi_Y10_outputs.txt", Y10)

#============================================#
# Open file to write results
f = open('SA_g6p_' + str(G6P) + '_n_' + str(n) \
         + '_Pi_' + str(Pi[0]) + '.txt', 'a+')

# Perform analysis Y2
print('Y2')
Si = sobol.analyze(problem, Y2, calc_second_order=False,\
                   print_to_console=True)

### Record the results to the file
f.write('For i = Y2:\n')
for key, value in Si.items():
    f.write('%s: %s\n' % (key, value))

f.write('\n')
f.write('=================================\n')

# Perform analysis Y4
print('Y4')
Si = sobol.analyze(problem, Y4, calc_second_order=False,\
                   print_to_console=True)

### Record the results to the file
f.write('For i = Y4:\n')
for key, value in Si.items():
    f.write('%s: %s\n' % (key, value))

f.write('\n')
f.write('================================\n')


# Perform analysis Y6
print('Y6')
Si = sobol.analyze(problem, Y6, calc_second_order=False,\
                   print_to_console=True)

### Record the results to the file
f.write('For i = Y6:\n')
for key, value in Si.items():
    f.write('%s: %s\n' % (key, value))

f.write('\n')
f.write('================================\n')

# Perform analysis Y8
print('Y8')
Si = sobol.analyze(problem, Y8, calc_second_order=False,\
                   print_to_console=True)

### Record the results to the file
f.write('For i = Y8:\n')
for key, value in Si.items():
    f.write('%s: %s\n' % (key, value))

f.write('\n')
f.write('================================\n')

# Perform analysis Y10
print('Y10')
Si = sobol.analyze(problem, Y10, calc_second_order=False,\
                   print_to_console=True)

### Record the results to the file
f.write('For i = Y10:\n')
for key, value in Si.items():
    f.write('%s: %s\n' % (key, value))

f.write('\n')
f.write('================================\n')

# Close result file #
f.close()

#=========================================================#
#==                THE END                              ==#
#=========================================================#
```

```
#=======================#
#       MAIN FILE       #
#=======================#

# Import some needed packages

from scipy.integrate import odeint
from scipy import integrate
import numpy as np
import matplotlib.pyplot as plt
from _model_ import *
#from conservation import *
#
#
# Define numerical solutions
# Step size
h = 0.01
#
# Solution
def solution(model, initial_cond, t0, t1, p):
    # p: parameters
    t = np.arange(t0, t1, h)
    sl = integrate.odeint(model, initial_cond, t, args=p)
    return(sl)

#
# Time grid
t0 = 0.0
t1 = 45.0
t = np.arange(t0, t1, h)

# Solutons to the full model
ys = []
yss = []

for i in range(len(y0)):
    ys.append(solution(MODEL, y0[i], t0, t1, p0))
    yss.append(solution(MODEL, y0[i], t0, t1, p0s))

ysa = []
for i in range(len(xl)):
    ysa.append(solution(MODEL, y0[1], t0, t1, p0l[i]))
#
#=====Conservation Test=====#
#plt.figure(figsize=(7, 5))
#plt.plot(t, glc, label='G')
#plt.plot(t, atp, label='ATP')
#plt.plot(t, Pi, label='Pi')
#plt.xlabel('Time ($s$)')
#plt.ylabel('$G6P$ concentration $mM$')


#==========================#
# Numerical solutions #
# and simplified model #

list_color = ['red', 'green', 'blue',\
              'indigo', 'crimson', 'black', 'maroon']

#==========================#
#
plt.figure(figsize=(7,5))
plt.plot(t, ys[0][:,3], color=list_color[0], linestyle='--',\
         label='$[P_i]$ = ' + str(Pi[0]) + ' $mM$')
plt.plot(t, yss[0][:,3], color=list_color[0], linestyle='-.', \
         label='$[P_i]$ = ' + str(Pi[0]) + ' $mM$')
for i in range(1, len(Pi)):
    plt.plot(t, ys[i][:,3], color=list_color[i], \
            label='$[P_i]$ = ' + str(Pi[i]) + ' $mM$')
    plt.plot(t, yss[i][:,3], color=list_color[i], \
            label='$[P_i]$ = ' + str(Pi[i]) + ' $mM$', \
            linestyle='-.')
plt.xlabel('Time ($s$)')
plt.ylabel('$G6P$ concentration $mM$')
plt.legend()

#=========================#
# SA illustration #

sa_color = ['g', 'r', 'b']
sa_labels = ['$0.7k_{-9}$', '$1.0k_{-9}$', '$1.3k_{-9}$']

fig=plt.figure(figsize=(7,5))
for i in range(len(xl)):
    plt.plot(t, ysa[i][:,3], color=sa_color[i],\
             label=sa_labels[i])
plt.xlabel('Time ($s$)')
plt.ylabel('$G6P$ concentration $mM$')
plt.legend()

plt.show()
#=======================================================#
#=                      END                             #
#=======================================================#
```

```
#======================#
#        MODEL         #
#======================#

def MODEL(y, t, k0, k1, k_1, k2, k_2, k3, k_3, k4, k_4, \

k5, k_5, k6, k_6, k7, k_7, k8, k_8, k9, k_9):
    #
    #
    # Define y

    _E_, _0_, _1_, _2_, _3_, _4_,\
        \
    _0E_,_1E_, _2E_, _3E_,\
        \
    _E0_, _E1_, _E2_, _E3_,\
        \
    _01E_, _02E_, _03E_,\
        \
    _0E0_, _0E1_, _0E2_, _0E3_,\
        \
    _1E0_, _1E1_, _1E2_, _1E3_,\
        \
    _2E0_, _2E1_, _2E3_,\
        \
    _3E0_, _3E1_, _3E2_, _3E3_,\
        \
    _E01_, _E02_, _E03_,\
        \
    _01E0_, _01E1_, _01E2_, _01E3_,\
        \
    _02E0_, _02E1_, _02E3_,\
        \
    _03E0_, _03E1_, _03E2_, _03E3_,\
        \
    _0E01_, _1E01_, _2E01_, _3E01_,\
        \
    _0E02_, _1E02_, _3E02_,\
        \
    _0E03_, _1E03_, _2E03_, _3E03_,\
        \
    _01E01_, _01E02_, _01E03_, _02E01_,\
        \
    _02E03_, _03E01_, _03E02_, _03E03_ = y
    #
    #
    # _E_: Hexokinase 1; _0_: Glucose
    # _1_: ATP; _2_: G6P; _3_: Pi; _4_: ADP
    # _xEy_: x, y substances bound at N, C domains, respectively.
    # Define dydt
    dydt=[]
    #1 Eq. for enyme _E_
    dydt.append(k0*_E01_ + k_1*_0E_ + k_2*_E0_ + k_3*_1E_ \

        + k_4*_E1_ + k_5*_2E_ + k_6*_E2_ + k_7*_3E_ + k_8*_E3_ \

        - _E_*((k1 + k2)*_0_ + (k3 + k4)*_1_ + (k5 + k6)*_2_ \

            + (k7 + k8)*_3_))

    #2 Eq. for _0_
    dydt.append(k_1*(_0E_ + _01E_ + _02E_ + _03E_ + _0E0_ \

            + _0E1_ + _0E2_ + _0E3_ + _01E0_ + _01E1_ + _01E2_ \

            + _01E3_ + _02E0_ + _02E1_ + _02E3_ + _03E0_ \

            + _03E1_ + _03E2_ + _03E3_ + _0E01_ + _0E02_ \
            + _0E03_ + _01E01_ + _01E02_ + _01E03_ + _02E01_ \
            + _02E03_ + _03E01_ + _03E02_ + _03E03_ )\
        + k_2*(_E0_ + _0E0_ + _1E0_ + _2E0_ + _3E0_ + _E01_ \
            + _E02_ + _E03_ + _01E0_ + _02E0_ + _03E0_ + _0E01_ \
            + _1E01_ + _2E01_ + _3E01_ +_0E02_ + _1E02_ + _3E02_ \
            + _0E03_ + _1E03_ + _2E03_ + _3E03_ + _01E01_ \
            + _01E02_ + _01E03_ + _02E01_ + _02E03_ + _03E01_ \
            + _03E02_ + _03E03_ )\
        - _0_*((k1 + k2)*(_E_ + _1E_ + _E1_ + _2E_ + _E2_ + _3E_ \
                + _E3_ + _1E1_ + _1E2_ + _1E3_ + _2E1_ + _2E3_ \
                + _3E1_ + _3E2_ + _3E3_)\
            + k1*(_E0_ + _1E0_ + _2E0_ + _3E0_ + _E01_ + _E02_ \
                + _E03_ + _1E01_ + _2E01_ + _3E01_ + _1E02_ \
                + _3E02_ + _1E03_ + _2E03_ + _3E03_) \
            + k2*(_0E_ + _01E_ + _02E_ + _03E_ + _0E1_ + _0E2_ \
                + _0E3_ + _01E1_ + _01E2_ + _01E3_ + _02E1_ \
                + _02E3_ + _03E1_ + _03E2_ + _03E3_)))
    #3 Eq. for _1_
    dydt.append(k_3*(_1E_ + _01E_ + _1E0_ + _1E1_ + _1E2_ + _1E3_ \
            + _01E0_ + _01E1_ + _01E2_ + _01E3_ + _1E01_ + _1E02_ \
            + _1E03_ + _01E01_ + _01E02_ + _01E03_) \
        + k_4*(_E1_ + _0E1_ + _1E1_ + _2E1_ + _3E1_ + _E01_ \
            + _01E1_ + _02E1_ + _03E1_ + _0E01_ + _1E01_ \
            + _2E01_ + _3E01_ + _01E01_ + _02E01_ + _03E01_) \
        - _1_*(k3*(_E_ + _0E_ + _E0_ + _E1_ + _E2_ + _E3_ + _0E0_ \
                + _0E1_ + _0E2_ + _0E3_ + _E01_ + _E02_ + _E03_ \
                + _0E01_ + _0E02_ + _0E03_) \
            + k4*(_E_ + _0E_ + _1E_ + _3E_ + _E0_ + _01E_ + _03E_ \
                + _0E0_ + _1E0_ + _3E0_ + _01E0_ + _03E0_)))
    #4 Eq. for _2_
    dydt.append(k0*(_E01_ + _0E01_ + _1E01_ + _2E01_ + _3E01_ \
                + _01E01_ + _02E01_ + _03E01_) \
            + k_5*(_2E_ + _02E_ + _2E0_ + _2E1_ + _2E3_ + _02E0_ \
                + _02E1_ + _02E3_ + _2E01_ + _2E03_ + _02E01_ \
                + _02E03_) \
            + k_6*(_E2_ + _0E2_ + _1E2_ + _E02_ + _01E2_ \
                + _0E02_ + _1E02_ + _01E02_) \

            + k_9*(_3E2_ + _03E2_ + _3E02_ +  _03E02_ ) \

        - _2_*(k5*(_E_ + _0E_ + _E0_ + _E1_ + _E3_ + _0E0_ \

                + _0E1_ + _0E3_ + _E01_ + _E03_ + _0E01_ + _0E03_) \

            + k6*(_E_ + _0E_ + _1E_ + _E0_ + _01E_ + _0E0_ \

                + _1E0_ + _01E0_) \

            + k9*(_3E_ + _03E_ + _3E0_ + _03E0_)))

    #5 Eq. for _3_
    dydt.append(k_7*(_3E_ + _03E_ + _3E0_ + _3E1_ + _3E2_ + _3E3_ \

            + _03E0_ + _03E1_ + _03E2_ + _03E3_ + _3E01_ \

            + _3E02_ + _3E03_ + _03E01_ + _03E02_ + _03E03_) \

        + k_8*(_E3_ + _0E3_ + _1E3_ + _2E3_ + _3E3_ + _E03_ \

            + _01E3_ + _02E3_ + _03E3_ + _0E03_ + _1E03_ \

            + _2E03_ + _3E03_ + _01E03_ + _02E03_ + _03E03_) \

        - _3_*(k7*(_E_ + _0E_ + _E0_ + _E1_ + _E2_ + _E3_ + _0E0_ \

                + _0E1_ + _0E2_ + _0E3_ + _E01_ + _E02_ + _E03_ \

                + _0E01_ + _0E02_ + _0E03_) \

            + k8*(_E_ + _0E_ + _1E_ + _3E_ + _E0_ + _01E_ + _03E_ \

                + _0E0_ + _1E0_ + _3E0_ + _01E0_ + _03E0_)))

    #6 Eq. for _4_
    dydt.append(k0*(_E01_ + _0E01_ + _1E01_ + _2E01_\

                + _3E01_ + _01E01_ + _02E01_ + _03E01_))
    #
    #
    #
    #7 Eq. for _0E_
    dydt.append(k0*_0E01_ + k1*_0_*_E_ + k_2*_0E0_ + k_3*_01E_ \

            + k_4*_0E1_ + k_5*_02E_ + k_6*_0E2_ + k_7*_03E_ \

            + k_8*_0E3_ \

        - _0E_*(k_1 + k2*_0_ + (k3 + k4)*_1_ + (k5 + k6)*_2_ \

            + (k7 + k8)*_3_))

    #8 Eq. for _1E_
    dydt.append(k0*_1E01_ + k_1*_01E_ + k_2*_1E0_ + k_4*_1E1_ \

            + k_6*_1E2_ + k_8*_1E3_ + _1_*k3*_E_ \

        - _1E_*(k_3 + (k1 + k2)*_0_ + k4*_1_ + k6*_2_ + k8*_3_))

    #9 Eq. for _2E_
    dydt.append(k0*_2E01_ + k_1*_02E_ + k_2*_2E0_ + k_4*_2E1_ \

            + k_8*_2E3_ + _2_*k5*_E_ \

        - _2E_*(k_5 + (k1 + k2)*_0_))
```

```
#10 Eq. for _3E_
dydt.append(k0*_3E01_ + k_1*_03E_ + k_2*_3E0_ + k_4*_3E1_ \

        + k_9*_3E2_ + k_8*_3E3_ + _3_*k7*_E_ \

    - _3E_*(k_7 + (k1 + k2)*_0_ + k4*_1_ + k9*_2_ + k8*_3_))

#11 Eq. for _E0_
dydt.append(k2*_E_*_0_ + k_1*_0E0_ + k_3*_1E0_ + k_5*_2E0_ \

        + k_7*_3E0_ + k_4*_E01_ + k_6*_E02_  + k_8*_E03_ \

    - _E0_*(k_2 + k1*_0_ + (k3 + k4)*_1_ + _2_*(k5 + k6) \

        + (k7 + k8)*_3_))

#12 Eq. for _E1_
dydt.append(k_1*_0E1_ + k_2*_E01_ + k_3*_1E1_ + k_5*_2E1_ \

        + k_7*_3E1_ + _1_*_E_*k4 \

    - _E1_*(k_4 + (k1 + k2)*_0_ + k3*_1_ + k5*_2_ + k7*_3_))

#13 Eq. for _E2_
dydt.append(k_1*_0E2_ + k_2*_E02_ + k_3*_1E2_ + k_7*_3E2_ \

        + _2_*k6*_E_ \

    -_E2_*(k_6 + (k1 + k2)*_0_ + k3*_1_ + k7*_3_))

#14 Eq. for _E3_
dydt.append(k_1*_0E3_ + k_2*_E03_ + k_3*_1E3_ + k_5*_2E3_ \

        + k_7*_3E3_ + _3_*k8*_E_ \

    - _E3_*(k_8 + (k1 + k2)*_0_ + k3 *_1_ + k5*_2_ + k7*_3_))

#
#
#15 Eq. for _01E_
dydt.append(k0*_01E01_ + k_2*_01E0_ + k_4*_01E1_ + k_6*_01E2_ \

        + k_8*_01E3_ + _0_*_1E_*k1 + _1_*k3*_0E_ \

    - _01E_*(k_1 + k_3 + k2*_0_ + k4*_1_ + k6*_2_ + k8*_3_))

#16 Eq. for _02E_
dydt.append(k0*_02E01_ + k_2*_02E0_ + k_4*_02E1_ + k_8*_02E3_ \

        + k1*_0_*_2E_ + _2_*k5*_0E_ \

    - _02E_*(k_1 + k_5 + k2*_0_))

#17 Eq. for _03E_
dydt.append(k0*_03E01_ + k_2*_03E0_ + k_4*_03E1_ + k_9*_03E2_ \

        + k_8*_03E3_ + _0_*_3E_*k1 + _3_*k7*_0E_ \

    - _03E_*(k_1 + k_7 + k2*_0_ + k4*_1_ + k9*_2_ + k8*_3_))

#
#18 Eq. for _0E0_
dydt.append(k_3*_01E0_ + k_4*_0E01_ + k_5*_02E0_ + k_6*_0E02_ \

        + k_7*_03E0_ + k_8*_0E03_ + _0_*(k1*_E0_ + k2*_0E_) \

    - _0E0_*(k_1 + k_2 + (k3 + k4)*_1_ + (k5 + k6)*_2_ \
        + (k7 + k8)*_3_))

#19 Eq. for _0E1_
dydt.append(k_2*_0E01_ + k_3*_01E1_ + k_5*_02E1_ + k_7*_03E1_ \

        + k1*_0_*_E1_ + _1_*k4*_0E_ \

    - _0E1_*(k_1 + k_4 + k2*_0_ + k3*_1_ + k5*_2_ + k7*_3_))

#20 Eq. for _0E2_
dydt.append(k_2*_0E02_ + k_3*_01E2_ + k_7*_03E2_ + k1*_0_*_E2_ \

        + _2_*k6*_0E_ \

    - _0E2_*(k_1 + k_6 + k2*_0_ + k3*_1_ + k7*_3_))

#21 Eq. for _0E3_
dydt.append(k_2*_0E03_ + k_3*_01E3_ + k_5*_02E3_ + k_7*_03E3_ \

        + k1*_0_*_E3_ + _3_*k8*_0E_ \

    - _0E3_*(k_1 + k_8 + k2*_0_ + k3*_1_ + k5*_2_ + k7*_3_))

#
#22 Eq. for _1E0_
dydt.append (k_1*_01E0_ + k_4*_1E01_ + k_6*_1E02_ \

        + k_8*_1E03_ + k2*_0_*_1E_ + _1_*k3*_E0_ \

    - _1E0_*(k_2 + k_3 + k1*_0_ + k4*_1_ + k6*_2_ + k8*_3_))

#23 Eq. for _1E1_
dydt.append(k_1*_01E1_ + k_2*_1E01_ + _1_*(k3*_E1_ + k4*_1E_) \

    - _1E1_*(k_3 + k_4 + (k1 + k2)*_0_))

#24 Eq. for _1E2_
dydt.append(k_1*_01E2_ + k_2*_1E02_ + _1_*k3*_E2_ + _2_*k6*_1E_ \

    - _1E2_*(k_3 + k_6 + (k1 + k2)*_0_))

#25 Eq. for _1E3_
dydt.append(k_1*_01E3_ + k_2*_1E03_ + _1_*k3*_E3_ + _3_*k8*_1E_ \

    - _1E3_*(k_3 + k_8 + (k1 + k2)*_0_))

#
#26 Eq. for _2E0_
dydt.append(k_1*_02E0_ + k_4*_2E01_ + k_8*_2E03_ \

        + k2*_0_*_2E_ + _2_*k5*_E0_ \

    - _2E0_*(k_2 + k_5 + k1*_0_))

#27 Eq. for _2E1_
dydt.append(k_1*_02E1_ + k_2*_2E01_ + _2_*k5*_E1_ \

    - _2E1_*(k_4 + k_5 + (k1 + k2)*_0_))

#28 Eq. for _2E3_
dydt.append(k_1*_02E3_ + k_2*_2E03_ + _2_*k5*_E3_ \

    - _2E3_*(k_5 + k_8 + (k1 + k2)*_0_))

#
#29 Eq. for _3E0_
dydt.append(k_1*_03E0_ + k_4*_3E01_ + k_9*_3E02_ \

        + k_8*_3E03_ + _0_*k2*_3E_ + _3_*k7*_E0_ \
    - _3E0_*(k_2 + k_7 + k1*_0_ + k4*_1_ + k9*_2_ + k8*_3_))

#30 Eq. for _3E1_
dydt.append(k_1*_03E1_ + k_2*_3E01_ + _1_*k4*_3E_ + _3_*k7*_E1_ \

    - _3E1_*(k_4 + k_7 + (k1 + k2)*_0_))

#31 Eq. for _3E2_
dydt.append(k_1*_03E2_ + k_2*_3E02_ + _2_*k9*_3E_ + _3_*k7*_E2_ \

    - _3E2_*(k_7 + k_9 + (k1 + k2)*_0_))

#32 Eq. for _3E3_
dydt.append(k_1*_03E3_ + k_2*_3E03_ + _3_*(k7*_E3_ + k8*_3E_) \

    - _3E3_*(k_7 + k_8 + (k1 + k2)*_0_))

#
#33 Eq. for _E01_
dydt.append(k_1*_0E01_ + k_3*_1E01_ + k_5*_2E01_ \

        + k_7*_3E01_ + _0_*k2*_E1_ + _1_*k4*_E0_ \

    - _E01_*(k0 + k_2 + k_4 + k1*_0_ + k3*_1_ + k5*_2_ + k7*_3_))

#34 Eq. for _E02_
dydt.append(k_1*_0E02_ + k_3*_1E02_ + k_7*_3E02_ \

        + _0_*k2*_E2_ + _2_*k6*_E0_ \

    - _E02_*(k_2 + k_6 + k1*_0_ + k3*_1_ + k7*_3_))

#35 Eq. for _E03_
dydt.append(k_1*_0E03_ + k_3*_1E03_ + k_5*_2E03_ \

        + k_7*_3E03_ + _0_*k2*_E3_ + _3_*k8*_E0_ \

    - _E03_*(k_2 + k_8 + _0_*k1 + _1_*k3 + _2_*k5 + _3_*k7))

#
#
#36 Eq. for _01E0_
dydt.append(k_4*_01E01_ + k_6*_01E02_ + k_8*_01E03_ \

        + _0_*(k1*_1E0_ + k2*_01E_) + _1_*k3*_0E0_ \

    - _01E0_*(k_1 + k_2 + k_3 + _1_*k4 + _2_*k6 + _3_*k8))

#37 Eq. for _01E1_
dydt.append(k_2*_01E01_ + _0_*k1*_1E1_ \

        + _1_*(k3*_0E1_ + k4*_01E_) \

    - _01E1_*(k_1 + k_3 + k_4 + _0_*k2))

#38 Eq. for _01E2_
dydt.append(k_2*_01E02_ + _0_*k1*_1E2_ + _1_*k3*_0E2_ \

        + _2_*k6*_01E_ \

    - _01E2_*(k_1 + k_3 + k_6 + _0_*k2))

#39 Eq. for _01E3_
dydt.append(k_2*_01E03_ + _0_*k1*_1E3_ \

        + _1_*k3*_0E3_ + _3_*k8*_01E_ \

    - _01E3_*(k_1 + k_3 + k_8 + _0_*k2 ))
```

```
#
#40 Eq. for _02E0_
dydt.append(k_4*_02E01_ + k_8*_02E03_ \

        + _0_*(k1*_2E0_ + k2*_02E_) + _2_*k5*_0E0_ \

    - _02E0_*(k_1 + k_2 + k_5))

#41 Eq. for _02E1_
dydt.append(k_2*_02E01_ + _0_*k1*_2E1_ + _2_*k5*_0E1_ \

    - _02E1_*(k_1 + k_4 + k_5 + _0_*k2))

#42 Eq. for _02E3_
dydt.append(k_2*_02E03_ + _0_*k1*_2E3_ + _2_*k5*_0E3_ \

    - _02E3_*(k_1 + k_5 + k_8 + _0_*k2))

#
#43 Eq. for _03E0_
dydt.append(k_4*_03E01_ + k_9*_03E02_ + k_8*_03E03_ \

        + _0_*(k1*_3E0_ + k2*_03E_) + _3_*k7*_0E0_ \

    - _03E0_*(k_1 + k_2 + k_7 + _1_*k4 + _2_*k9 + _3_*k8))

#44 Eq. for _03E1_
dydt.append(k_2*_03E01_ + _0_*k1*_3E1_ + _1_*k4*_03E_ \

        + _3_*k7*_0E1_ \

    - _03E1_*(k_1 + k_4 + k_7 + _0_*k2))

#45 Eq. for _03E2_
dydt.append(k_2*_03E02_ + _0_*k1*_3E2_ \

        + _2_*k9*_03E_ + _3_*k7*_0E2_ \

    - _03E2_*(k_1 + k_9 + k_7 + _0_*k2))

#46 Eq. for _03E3_
dydt.append(k_2*_03E03_ + _0_*k1*_3E3_ \

        + _3_*(k7*_0E3_ + k8*_03E_) \

    - _03E3_*(k_1 + k_7 + k_8 + _0_*k2))

#
#47 Eq. for _0E01_
dydt.append(k_3*_01E01_ + k_5*_02E01_ + k_7*_03E01_ \

        + _0_*(k1*_E01_ + k2*_0E1_) + _1_*k4*_0E0_ \

    - _0E01_*(k0 + k_1 + k_2 + k_4 + _1_*k3 + _2_*k5 + _3_*k7))

#48 Eq. for _1E01_
dydt.append(k_1*_01E01_ + _0_*k2*_1E1_ \

        + _1_*(k3*_E01_ + k4*_1E0_) \

    - _1E01_*(k0 + k_2 + k_3 + k_4 + _0_*k1))

#49 Eq. for _2E01_
dydt.append(k_1*_02E01_ + _0_*k2*_2E1_ + _2_*k5*_E01_ \

    - _2E01_*(k0 + k_2 + k_4 + k_5 + _0_*k1))

#50 Eq. for _3E01_
```

```
dydt.append(k_1*_03E01_ + _0_*k2*_3E1_ + _1_*k4*_3E0_ \

        + _3_*k7*_E01_ \

    - _3E01_*(k0 + k_2 + k_4 + k_7 + _0_*k1))

#
#51 Eq. for _0E02_
dydt.append(k_3*_01E02_ + k_7*_03E02_

        + _0_*(k1*_E02_ + k2*_0E2_) + _2_*k6*_0E0_ \

    - _0E02_*(k_1 + k_2 + k_6 + _1_*k3 + _3_*k7))

#52 Eq. for _1E02_
dydt.append(k_1*_01E02_ + _0_*k2*_1E2_ + _1_*k3*_E02_ \

        + _2_*k6*_1E0_ \

    - _1E02_*(k_2 + k_3 + k_6 + _0_*k1))

#53 Eq. for _3E02_
dydt.append(k_1*_03E02_ + _0_*k2*_3E2_ + _2_*k9*_3E0_ \

        + _3_*k7*_E02_ \

    - _3E02_*(k_2 + k_9 + k_7 + _0_*k1))

#
#54 Eq. for _0E03_
dydt.append(k_3*_01E03_ + k_5*_02E03_ + k_7*_03E03_ \

        + _0_*(k1*_E03_ + k2*_0E3_) + _3_*k8*_0E0_ \

    - _0E03_*(k_1 + k_2 + k_8 + _1_*k3 + _2_*k5 + _3_*k7))

#55 Eq. for _1E03_
dydt.append(k_1*_01E03_ + _0_*k2*_1E3_ + _1_*k3*_E03_ \

        + _3_*k8*_1E0_ \

    - _1E03_*(k_2 + k_3 + k_8 + _0_*k1))

#56 Eq. for _2E03_
dydt.append(k_1*_02E03_ + _0_*k2*_2E3_ + _2_*k5*_E03_ \

    - _2E03_*(k_2 + k_5 + k_8 + _0_*k1))

#57 Eq. for _3E03_
dydt.append(k_1*_03E03_ + _0_*k2*_3E3_ \

        + _3_*(k7*_E03_ + k8*_3E0_) \

    - _3E03_*(k_2 + k_7 + k_8 + _0_*k1))

#
#
#58 Eq. for _01E01_
dydt.append(_0_*(k1*_1E01_ + k2*_01E1_) \

        + _1_*(k3*_0E01_ + k4*_01E0_) \

    - _01E01_*(k0 + k_1 + k_2 + k_3 + k_4))

#59 Eq. for _01E02_
dydt.append(_0_*(k1*_1E02_ + k2*_01E2_) \

        + _1_*k3*_0E02_ + _2_*k6*_01E0_ \
        - _01E02_*(k_1 + k_2 + k_3 + k_6))

    #60 Eq. for _01E03_
    dydt.append(_0_*(k1*_1E03_ + k2*_01E3_) \

            + _1_*k3*_0E03_  + _3_*k8*_01E0_ \

        - _01E03_*(k_1 + k_2 + k_3 + k_8))

    #
    #61 Eq. for _02E01_
    dydt.append(_0_*(k1*_2E01_ + k2*_02E1_) + _2_*k5*_0E01_ \

        - _02E01_*(k0 + k_1 + k_2 + k_4 + k_5))

    #62 Eq. for _02E03_
    dydt.append(_0_*(k1*_2E03_ + k2*_02E3_) + _2_*k5*_0E03_ \

        - _02E03_*(k_1 + k_2 + k_5 + k_8))

    #
    #63 Eq. for _03E01_
    dydt.append(_0_*(k1*_3E01_ + k2*_03E1_) \

            + _1_*k4*_03E0_ + _3_*k7*_0E01_ \

        - _03E01_*(k0 + k_1 + k_2 + k_4 + k_7))

    #64 Eq. for _03E02_
    dydt.append(_0_*(k1*_3E02_ + k2*_03E2_) \

            + _2_*k9*_03E0_ + _3_*k7*_0E02_ \

        - _03E02_*(k_1 + k_2 + k_9 + k_7))

    #65 Eq. for _03E03_
    dydt.append(_0_*(k1*_3E03_ + k2*_03E3_) \

            + _3_*(k7*_0E03_ + k8*_03E0_) \

        - _03E03_*(k_1 + k_2 + k_7 + k_8))

    #--------------#
    # Return dydt  #
    #--------------#
    return(dydt)
#


# -------------------#
# INITIAL CONDITIONS #
#     Unit: mM       #
#--------------------#

# E, 0, 1, 2, 3, 4
# <=> HK, G, ATP, G6P, Pi, ADP
#
E, G, ATP = 6.65e-2, 2.5, 3.0
#
G6P, ADP = 0.0, 0.0

#Pi = [0.0, 2.0, 10.0]
Pi = [0.0, 1.0, 2.0, 4.0, 6.0, 10.0, 15.0]

#G6P = [0.0, 1.0, 2.0, 3.0]

# Pi = 6.0 should be changed to 8.0
y0 = []
```

```
for y in Pi:
    y0.append([E, G, ATP, G6P, y, ADP,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0, 0.0,\
     0.0, 0.0, 0.0, 0.0, 0.0])

#print(y0)
#
#
#===================================#
# PARAMETER VALUES FOR SIMULATIONS  #
#===================================#

##------------------##
## FROMM MODEL      ##
##==================##
k0 = 63.0

KmG = 0.053

a = 99.0
# For Glucose
k1, k_1 = (a + 1.0)*k0/KmG, a*k0

k2, k_2 = k1, k_1

# For ATP
# Km
KmA = 0.7
k3, k_3 = (a + 1.0)*k0/KmA, a*k0

# Ki =
k4, k_4 = k3, k_3

# For G6P
c = 1.0e-0

# Ki = 0.71 mM (N)
k5, k_5 = c*k3, 0.71*k3*c

# Ki = 54 microM = 0.054 mM(C)
k6, k_6 = c*k3, 0.054*k3*c

# For Pi
# K_i = 0.022mM
k7, k_7 = k3, 0.022*k3

# Ki = 0.22mM
k8, k_8 = k3, 0.22*k3

# For G6P binding to _3E_, _03E_, and _03E0_
u = 0.10
v = 1.0e-1
k9, k_9 = c*v*k3, c*v*u*k3


p0 = (k0, k1, k_1, k2, k_2, k3, k_3, k4, k_4, \
         k5, k_5, k6, k_6, k7, k_7, k8, k_8, k9, k_9)

for i in range(len(p0)):
    print(p0[i])
#==Simplified model==#
```

```
z = 0.0
p0s =(k0, z*k1, z*k_1, k2, k_2, z*k3, z*k_3, k4, k_4, \
      k5, k_5, k6, k_6, k7, k_7, k8, k_8, k9, k_9)

# ==SA illustration ==#
p0l = []
x = 0.3
xl = [1.0 - x, 1.0, 1.0 + x]

for i in xl:
    p0l.append((k0, k1, k_1, k2, k_2, k3, k_3, k4, k_4, \
             k5, k_5, k6, k_6, k7, k_7, k8, k_8, k9, i*k_9))
```

Bibliography

[1] E Bartholomeus Kuettner, Karina Kettner, Antje Keim, Dmitri I Svergun, Daniela Volke, David Singer, Ralf Hoffmann, Eva-Christina Müller, Albrecht Otto, Thomas M Kriegel, et al. Crystal structure of hexokinase klhxk1 of kluyveromyces lactis a molecular basis for understanding the control of yeast hexokinase functions via covalent modification and oligomerization. *Journal of Biological Chemistry*, 285(52):41019–41033, 2010.

[2] Alexander S Rose, Anthony R Bradley, Yana Valasatava, Jose M Duarte, Andreas Prlić, and Peter W Rose. Ngl viewer: web-based molecular graphics for large complexes. *Bioinformatics*, 34(21):3755–3758, 2018.

[3] Peter K Robinson. Enzymes: principles and biotechnological applications. *Essays in biochemistry*, 59:1–41, 2015.

[4] Mcat biochemistry review. `https://schoolbag.info/chemistry/mcat_biochemistry/10.html`. Accessed: 2019- 9 -23.

[5] Warren Lyford DeLano. Pymol by schrödinger. `https://pymol.org/2/`. Accessed: 2019 - 10 - 08.

[6] Hyaluroniase - wikipedia. `https://en.wikipedia.org/wiki/Hyaluronidase`. Accessed: 2019 - 9 - 24.

[7] M. J. Jedrzejas, L. V. Mello, B. L. de Groot, and S. Li. Mechanism of hyaluronan degradation by streptococcus pneumoniae hyaluronate lyase. *The Journal of Biological Chemistry*, 277:28287–28297, 2002.

[8] Enzyme nomenclature. `https://www.qmul.ac.uk/sbcs/iubmb/enzyme/`. Accessed: 2019 - 9 - 26.

[9] Enzyme definition and classification. `https://www.creative-enzymes.com/resource/enzyme-definition-and-classification_18.html`. Accessed: 2019 - 9 - 26.

[10] Jordan Chapman, Ahmed Ismail, and Cerasela Dinu. Industrial applications of enzymes: Recent advances, techniques, and outlooks. *Catalysts*, 8(6):238, 2018.

[11] M. M. Rapport, A. Linker, and K. Meyer. The hydrolysis of hyaluronic acid by pneumococcal hyaluronidase. *Journal of Biological Chemistry*, 192(1):283–291, 1951.

[12] Anselme Payen and Jean-François Persoz. Mémoire sur la diastase, les principaux produits de ses réactions, et leurs applications aux arts industriels. *Ann. chim. phys*, 53:73–92, 1833.

[13] Wilhelm Kühne. Ueber das verhalten verschiedener organisirter und sog. ungeformter fermente. *Naturhistorisch-medicinischen vereins, Heidelberg*, 1877. Also available as https://archive.org/details/verhandlungendes7477natu/page/190.

[14] Perry A Frey and Adrian D Hegeman. *Enzymatic reaction mechanisms.* Oxford University Press, 2007.

[15] Introduction to enzymes. http://www.worthington-biochem.com/introBiochem/ introEnzymes.html. Accessed: 2019-5-01.

[16] Tim DH Bugg. *Introduction to enzyme and coenzyme chemistry.* John Wiley & Sons, 2012.

[17] Athel Cornish-Bowden and Athel Cornish-Bowden. *Fundamentals of enzyme kinetics*, volume 510. Wiley-Blackwell Weinheim, Germany, 2012.

[18] K.B. Taylor. *Enzyme kinetics and mechanisms.* Kluwer Academic Publishers, Dordrecht, 2004.

[19] Albert L Lehninger, David L Nelson, Michael M Cox, Michael M Cox, et al. *Lehninger principles of biochemistry.* Macmillan, 2005.

[20] Anna Radzicka and Richard Wolfenden. A proficient enzyme. *Science*, 267(5194):90–93, 1995.

[21] Enzyme faq. https://enzyscience.com/pages/enzyme-faq. Accessed: 2019 - 9 - 24.

[22] BRENDA. Information on ec 7.2.2.1 - na+-transporting two-sector atpase. https://www.brenda-enzymes.org/enzyme.php?ecno=7.2.2.1. Accessed: 2019 - 10 - 10.

[23] BRENDA. Information on ec 7.4.2.1 - abc-type polar-amino-acid transporter. https://www.brenda-enzymes.org/enzyme.php?ecno=7.4.2.1. Accessed: 2019 - 10 - 10.

[24] Enzyme commission number. https://www.abbexa.com/enzyme-commission-number. Accessed: 2019 - 9 - 24.

[25] Brenda - information on ec 4.2.2.1 - hyaluronate lyase. https://www.brenda-enzymes.org/enzyme.php?ecno=4.2.2.1. Accessed: 2019 - 9 - 24.

[26] Graham L Pettipher and Malcolm J Latham. Characteristics of enzymes produced by ruminococcus flavefaciens which degrade plant cell walls. *Microbiology*, 110(1):21–27, 1979.

[27] Inger Johansson, Gunilla Ekstroem, Bob Scholte, David Puzycki, Hans Jörnvall, and Magnus Ingelman-Sundberg. Ethanol-, fasting-, and aceton-inducible cytochromes p-450 in rat liver: regulation and characteristics of enzymes belonging to the iib and iie gene subfamilies. *Biochemistry*, 27(6):1925–1934, 1988.

[28] Ole Kirk, Torben Vedel Borchert, and Claus Crone Fuglsang. Industrial enzyme applications. *Current opinion in biotechnology*, 13(4):345–351, 2002.

[29] Shosuke Okamoto, Akiko Hijikata-Okunomiya, Keiko Wanaka, Yoshio Okada, and Utako Okamoto. Enzyme-controlling medicines: introduction. In *Seminars in thrombosis and hemostasis*, volume 23, pages 493–501. Copyright© 1997 by Thieme Medical Publishers, Inc., 1997.

[30] Karlheinz Drauz, Harald Gröger, and Oliver May. *Enzyme Catalysis in Organic Synthesis, 3 Volume Set*, volume 1. John Wiley & Sons, 2012.

[31] W Saenger, M Noltemeyer, PC Manor, B Hingerty, and B Klar. "induced-fit"-type complex formation of the model enzyme α-cyclodextrin. *Bioorganic Chemistry*, 5(2):187–195, 1976.

[32] Roy M Daniel, Rachel V Dunn, John L Finney, and Jeremy C Smith. The role of dynamics in enzyme activity. *Annual review of biophysics and biomolecular structure*, 32(1):69–92, 2003.

[33] James P Keener and James Sneyd. *Mathematical physiology*, volume 1. Springer, 1998.

[34] WW Cleland. The kinetics of enzyme-catalyzed reactions with two or more substrates or products: Ii. inhibition: Nomenclature and theory. *Biochimica et Biophysica Acta (BBA)-Specialized Section on Enzymological Subjects*, 67:173–187, 1963.

[35] Brenda - information on ec 2.7.1.1 - hexkinase. `https://www.brenda-enzymes.org/enzyme.php?ecno=2.7.1.1`. Accessed: 2019 - 9 - 24.

[36] Haike Ghazarian, Brian Idoni, and Steven B Oppenheimer. A glycobiology review: carbohydrates, lectins and implications in cancer therapeutics. *Acta histochemica*, 113(3):236–247, 2011.

[37] LG Wade. Organic chemistry prentice-hall. *New Jersey*, 1995.

[38] Antoinette C O'sullivan. Cellulose: the structure slowly unravels. *Cellulose*, 4(3):173–207, 1997.

[39] K. Meyer A. Linker and P. Hoffman. The production of unsaturated uronides by bacterial hyaluronidases. *The Journal of Biological Chemistry*, 221:1109, 1955.

[40] J. Necas, L. Bartosikova, P. Brauner, and J. Kolar. Hyaluronic acid (hyaluronan): a review. *Veterinarni Medicina*, 53(8):397–411, 2008.

[41] R.V. Stick and S.J. Williams. *Carbohydrates: The Essential Molecules of Life*. Elsevier, 2009.

[42] Michael Sinnott. *Carbohydrate chemistry and biochemistry: structure and mechanism*. Royal Society of Chemistry, 2007.

[43] Philip C Calder. Glycogen structure and biogenesis. *International Journal of Biochemistry*, 23(12):1335–1352, 1991.

[44] Carbohydrate-active enzymes. https://www.cazypedia.org/index.php/Carbohydrate-active_enzymes. Accessed: 2019-9-11.

[45] James A Campbell, Gideon J Davies, Vincent Bulone, and Bernard Henrissat. A classification of nucleotide-diphospho-sugar glycosyltransferases based on amino acid sequence similarities. *Biochemical Journal*, 326(Pt 3):929, 1997.

[46] Pedro M Coutinho, Emeline Deleury, Gideon J Davies, and Bernard Henrissat. An evolving hierarchical family classification for glycosyltransferases. *Journal of molecular biology*, 328(2):307–317, 2003.

[47] Pedro M Coutinho, Corinne Rancurel, Mark Stam, Thomas Bernard, Francisco M Couto, Etienne GJ Danchin, and Bernard Henrissat. Carbohydrate-active enzymes database: principles and classification of glycosyltransferases. *Bioinformatics for Glycobiology and Glyconomics: An Introduction. West Sussex: John Wiley & Sons*, pages 91–118, 2009.

[48] Bernard Henrissat. A classification of glycosyl hydrolases based on amino acid sequence similarities. *Biochemical journal*, 280(2):309–316, 1991.

[49] Bernard Henrissat and Amos Bairoch. New families in the classification of glycosyl hydrolases based on amino acid sequence similarities. *Biochemical journal*, 293(3):781–788, 1993.

[50] Bernard Henrissat and Amos Bairoch. Updating the sequence-based classification of glycosyl hydrolases. *Biochemical Journal*, 316(Pt 2):695, 1996.

[51] Vincent Lombard, Thomas Bernard, Corinne Rancurel, Harry Brumer, Pedro M Coutinho, and Bernard Henrissat. A hierarchical classification of polysaccharide lyases for glycogenomics. *Biochemical Journal*, 432(3):437–444, 2010.

[52] Marie-Line Garron and Miroslaw Cygler. Structural and mechanistic classification of uronic acid-containing polysaccharide lyases. *Glycobiology*, 20(12):1547–1573, 2010.

[53] Gideon J Davies, Tracey M Gloster, and Bernard Henrissat. Recent structural insights into the expanding world of carbohydrate-active enzymes. *Current opinion in structural biology*, 15(6):637–645, 2005.

[54] Peter Biely. Microbial carbohydrate esterases deacetylating plant polysaccharides. *Biotechnology advances*, 30(6):1575–1588, 2012.

[55] Anthony Levasseur, Elodie Drula, Vincent Lombard, Pedro M Coutinho, and Bernard Henrissat. Expansion of the enzymatic repertoire of the cazy database to integrate auxiliary redox enzymes. *Biotechnology for biofuels*, 6(1):41, 2013.

[56] Brandi L Cantarel, Pedro M Coutinho, Corinne Rancurel, Thomas Bernard, Vincent Lombard, and Bernard Henrissat. The carbohydrate-active enzymes database (cazy): an expert resource for glycogenomics. *Nucleic acids research*, 37(suppl_1):D233–D238, 2008.

[57] Mark H Holmes. *Introduction to the foundations of applied mathematics*, volume 56. Springer Science & Business Media, 2009.

[58] Edda Klipp, Wolfram Liebermeister, Christoph Wierling, and Axel Kowald. *Systems biology: a textbook*. John Wiley & Sons, 2016.

[59] Eduardo D Sontag. Lecture notes on mathematical systems biology. https://www.math.rutgers.edu/docman-lister/math-main/academics/undergraduate/interdisciplinary-majors/biomathematics-interdisciplinary-major/2198-lecture-notes-on-mathematical-systems-biology/file, 2015. Accessed: 2019-8-30.

[60] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[61] Fuchang Gao and Lixing Han. Implementing the nelder-mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1):259–277, 2012.

[62] Dieter Kraft. A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[63] G Peckham. A new method for minimising a sum of squares without calculating gradients. *The Computer Journal*, 13(4):418–420, 1970.

[64] Joseph-Frédéric Bonnans, Jean Charles Gilbert, Claude Lemaréchal, and Claudia A Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006.

[65] Minimization of scalar function of one or more variables. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html`. Accessed: 2019-09-20.

[66] Singiresu S Rao. Optimization theory and applications. *JOHN WILEY & SONS, INC., 605 THIRD AVE., NEW YORK, NY 10158, USA, 1983, 550*, 1983.

[67] Yurij G Evtushenko and J Stoer. *Numerical optimization techniques.* Optimization Software, Incorporated, Publications Division, 1985.

[68] SciPy. Scipy open source python library. `https://www.scipy.org/`, 2001. Accessed: 2019-4-19.

[69] Xiaohong Chen, Han Hong, and Elie Tamer. Measurement error models with auxiliary data. *The Review of Economic Studies*, 72(2):343–366, 2005.

[70] Leonard A Stefanski and Raymond J Carroll. Conditional scores and optimal scores for generalized linear measurement-error models. *Biometrika*, 74(4):703–716, 1987.

[71] John P Buonaccorsi. *Measurement error: models, methods, and applications.* Chapman and Hall/CRC, 2010.

[72] Matthew Blackwell, James Honaker, and Gary King. A unified approach to measurement error and missing data: overview and applications. *Sociological Methods & Research*, 46(3):303–341, 2017.

[73] Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1):271–280, 2001.

[74] Andrea Saltelli, Stefano Tarantola, Francesca Campolongo, and Marco Ratto. Sensitivity analysis in practice: a guide to assessing scientific models. *Chichester, England*, 2004.

[75] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis: the primer.* John Wiley & Sons, 2008.

[76] Sergei Kucherenko, Stefano Tarantola, and Paola Annoni. Estimation of global sensitivity indices for models with dependent variables. *Computer Physics Communications*, 183(4):937–946, 2012.

[77] Francesca Campolongo, Jessica Cariboni, and Andrea Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental modelling & software*, 22(10):1509–1518, 2007.

[78] Andrea Saltelli, Stefano Tarantola, and KP-S Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.

[79] RI Cukier, CM Fortuin, Kurt E Shuler, AG Petschek, and JH Schaibly. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i theory. *The Journal of chemical physics*, 59(8):3873–3878, 1973.

[80] S Kucherenko and IM Sobol. Derivative based global sensitivity measures and their link with global sensitivity indices. *Mathematics and Computers in Simulation*, 79(10):3009–3017, 2009.

[81] GEB Archer, Andrea Saltelli, and IM Sobol. Sensitivity measures, anova-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation*, 58(2):99–120, 1997.

[82] X-Y Zhang, MN Trame, LJ Lesko, and S Schmidt. Sobol sensitivity analysis: a tool to guide the development and evaluation of systems pharmacology models. *CPT: pharmacometrics & systems pharmacology*, 4(2):69–79, 2015.

[83] Jon Herman and Will Usher. Salib: an open-source python library for sensitivity analysis. *The Journal of Open Source Software*, 2(9), 2017.

[84] Python. Python software foundation. `https://www.python.org/`, 1994. Accessed: 2019-4-19.

[85] Guido van Rossum. The history of python. `https://python-history.blogspot.com/2009/01/brief-timeline-of-python.html`. Accessed: 2019 - 9 - 22.

[86] History and license. `https://docs.python.org/3/license.html`. Accessed:2019 - 9 - 22.

[87] K Jarrod Millman and Michael Aivazis. Python for scientists and engineers. *Computing in Science & Engineering*, 13(2):9–12, 2011.

[88] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22, 2011.

[89] Numpy. `https://numpy.org/`. Accessed: 2019 - 9 - 22.

[90] Travis E. Oliphant. Guide to numpy. `http://csc.ucdavis.edu/~chaos/courses/nlp/Software/NumPyBook.pdf`, Dec 7, 2006. Accessed: 2019 - 9 - 22.

[91] History of scipy. `https://scipy.github.io/old-wiki/pages/History_of_SciPy`. Accessed: 2019 - 9 - 22.

[92] LSODA. The ldosa program in the fortran library odepack. http://www.oecd-nea.org/tools/abstract/detail/uscd1227, -. Accessed: 2019-4-19.

[93] Stiff systems. http://www.scholarpedia.org/article/Stiff_systems. Accessed: 2019 - 9 - 22.

[94] Gerhard Wanner and Ernst Hairer. *Solving ordinary differential equations II.* Springer Berlin Heidelberg, 1996.

[95] John Denholm Lambert. *Numerical methods for ordinary differential systems: the initial value problem.* John Wiley & Sons, Inc., 1991.

[96] François E Cellier and Ernesto Kofman. *Continuous system simulation.* Springer Science & Business Media, 2006.

[97] Optimization (scipy.optimize). https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html. Accessed: 2019 - 9 - 23.

[98] Basics - salib 1.3.8. https://salib.readthedocs.io/en/latest/basics.html#what-is-sensitivity-analysis. Accessed: 2019 - 10 - 04.

[99] Andrea Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer physics communications*, 145(2):280–297, 2002.

[100] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259–270, 2010.

[101] Max D Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.

[102] Matplotlib. https://matplotlib.org/. Accessed: 2019 - 9 - 22.

[103] Matplotlib - wikipedia. https://en.wikipedia.org/wiki/Matplotlib. Accessed: 2019 - 9 - 23.

[104] J Whitaker. The matplotlib basemap toolkit user's guide. *Matplotlib Basemap Toolkit documentation, February*, 2011.

[105] Welcome to the matplotlib basemap toolkit documentation. https://matplotlib.org/basemap/index.html. Accessed: 2019 - 9 - 23.

[106] Vinh Q Mai, Tuoi T Vo, and Martin Meere. Modelling hyaluronan degradation by streptococcus pneumoniae hyaluronate lyase. *Mathematical biosciences*, 303:126–138, 2018.

[107] K. Meyer and J. W. Palmer. The polysaccharide of the vitreous humor. *Journal of Biological Chemistry*, 107(3):629–634, 1934.

[108] J. R. E. Fraser, T. C. Laurent, and U. B. G. Laurent. Hyaluronan: its nature, distribution, functions and turnover. *Journal of Internal Medicine*, 242(1):27–33, 1997.

[109] A. Fakhari and C. Berkland. Applications and emerging trends of hyaluronic acid in tissue engineering, as a dermal filler and in osteoarthritis treatment. *Acta Biomaterialia*, 9(7):7081 – 7092, 2013.

[110] R. Stern, A. A. Asari, and K. N. Sugahara. Hyaluronan fragments: An information-rich system. *European Journal of Cell Biology*, 85:699–715, 2006.

[111] R. N. Feinberg and D. C. Beebe. Hyaluronate in vasculogenesis. *Science*, 220(4602):1177–1179, 1983.

[112] J. V. Forrester and E. A. Balazs. Inhibition of phagocytosis by high molecular weight hyaluronate. *Immunology*, 40:435–446, 1980.

[113] Hans-Joachim Lueke and P. Prehm. Synthesis and shedding of hyaluronan from plasma membranes of human fibroblasts and metastatic and non-metastatic melanoma cells. *Biochemical Journal*, 343(1):71–75, 1999.

[114] J. M. Delmage, D. R. Powars, P. K. Jaynes, and S. E. Allerton. The selective suppression of immunogenicity by hyaluronic acid. *Annals of Clinical & Laboratory Science*, 16(4):303–310, 1986.

[115] Kazuki N. Sugahara, Toshiyuki Murai, Hitomi Nishinakamura, Hiroto Kawashima, Hideyuki Saya, and Masayuki Miyasaka. Hyaluronan oligosaccharides induce cd44 cleavage and promote cell migration in cd44-expressing tumor cells. *Journal of Biological Chemistry*, 278(34):32259–32265, 2003.

[116] S. Gariboldi, M. Palazzo, L. Zanobbio, S. Selleri, M. Sommariva, L. Sfondrini, S. Cavicchini, A. Balsari, and C. Rumio. Low molecular weight hyaluronic acid increases the self-defense of skin epithelium by induction of β-defensin 2 via tlr2 and tlr4. *The Journal of Immunology*, 181(3):2103–2110, 2008.

[117] N. S. El-Safory, A. E. Fazary, and C. K. Lee. Hyaluronidase, a group of glycosidases: Current and future perspectives. *Carbohydrate Polymers*, 81:165–181, 2010.

[118] R. Stern, G. Kogan, M. J. Jedrzejas, and L. Šoltés. The many ways to cleave hyaluronan. *Biotechnology Advances*, 25(6):537 – 557, 2007.

[119] S. Dong J. R. Baker and D. G. Pritchard. The hyaluronan lyase of streptococcus pyogenes bacteriophage h4489a. *Biochemical Journal*, 365:317–322, 2002.

[120] E. J. Menzel and C. Farr. Hyaluronidase and its substrate hyaluronan: biochemistry, biological activities and therapeutic uses. *Cancer Letters*, 131(1):3 – 11, 1998.

[121] Graham J Boulnois. Pneumococcal proteins and the pathogenesis of disease caused by streptococcus pneumoniae. *Microbiology*, 138(2):249–259, 1992.

[122] J. H. Humphrey. Hyaluronidase production by pneumococci. *The Journal of Pathology and Bacteriology*, 56(2):273–275, 1944.

[123] Md Sohail Akhtar and Vinod Bhakuni. Streptococcus pncumoniae hyaluronate lyase: An overview. *Current Science*, 86(2), 2004.

[124] M. J. Jedrzejas, R. B. Mewbourne, L. Chantalat, and D. T. McPherson. Expression and purification of streptococcus pneumoniae hyaluronate lyase from escherichia coli. *Protein Expression and Purification*, 13(1):83 – 89, 1998.

[125] Stephen J Kelly, Kenneth B Taylor, Songlin Li, and Mark J Jedrzejas. Kinetic properties of streptococcus pneumoniae hyaluronate lyase. *Glycobiology*, 11(4):297–304, 2001.

[126] Karthe Ponnuraj and Mark J Jedrzejas. Mechanism of hyaluronan binding and degradation: structure of streptococcus pneumoniae hyaluronate lyase in complex with hyaluronic acid disaccharide at 1.7 Å resolution. *Journal of molecular biology*, 299(4):885–895, 2000.

[127] D. G. Pritchard, B. Lin, T. R. Willingham, and J. R. Baker. Characterization of the group b streptococcal hyaluronate lyase. *Archives of Biochemistry and Biophysics*, 315(2):431 – 437, 1994.

[128] Hongxing Niu, Nilay Shah, and Cleo Kontoravdi. Modelling of amorphous cellulose depolymerisation by cellulases, parametric studies and optimisation. *Biochemical Engineering Journal*, 105:455–472, 2016.

[129] Seth E. Levine, Jerome M. Fox, Harvey W. Blanch, and Douglas S. Clark. A mechanistic model of the enzymatic hydrolysis of cellulose. *Biotechnology and Bioengineering*, 107(1):37–51, 2010.

[130] Andrea Asztalos, Marcus Daniels, Anurag Sethi, Tongye Shen, Paul Langan, Antonio Redondo, and Sandrasegaram Gnanakaran. A coarse-grained model for synergistic action of multiple enzymes on cellulose. *Biotechnology for Biofuels*, 5(55), 2012.

[131] A. L. Lucius, N. K. Maluf, C. J. Fischer, and T. M. Lohman. General methods for analysis of sequential n-step kinetic mechanisms: application to single turnover kinetics helicase-catalyzed dna unwinding. *Biophysical Journal*, 84:2224–2239, 2003.

[132] E. Praestgaard, J. Elmerdahl, L. Murphy, S. Nymand, K. C. McFarland, K. Borch, and Peter Westh. A kinetic model for the burst phase of processive cellulases. *The FEBS Journal*, 278:1547–1560, 2011.

[133] N. Cruys-Bagger, J. Elmerdahl, E. Praestgaard, H. Tatsumi, N. Spodsberg, K. Borch, and P. Westh. Pre-steady-state kinetics for hydrolysis of insoluble cellulose by cellobiohydrolase cel7a. *The Journal of Biological Chemistry*, 287(22):18451–18458, 2012.

[134] N. Cruys-Bagger, J. Elmerdahl, E. Praestgaard, K. Borch, and P. Westh. A steady-state theory for processive cellulases. *The FEBS Journal*, 280:3952–3961, 2013.

[135] Python software foundation. `https://www.python.org/`. Accessed: 2017-12-09.

[136] Scipy open source python library. `https://www.scipy.org/`. Accessed: 2017-12-09.

[137] The `odeint` solver in the `integrate` module of the scipy library. `https://docs.scipy.org/doc/scipy-0.19.1/reference/generated/scipy.integrate.odeint.html`. Accessed: 2017-12-09.

[138] The `ldosa` program in the fortran library `odepack`. `http://www.oecd-nea.org/tools/abstract/detail/uscd1227`. Accessed: 2017-12-09.

[139] The `minimize` routine in the `optimize` module of the scipy library. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html`. Accessed: 2017-12-12.

[140] The sequential least square programming method. `https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html#optimize-minimize-slsqp`. Accessed: 2017-12-12.

[141] Sensitivity analysis library for systems modeling. `https://github.com/SALib`. Accessed: 2017-12-14.

[142] Premomoy Ghosh. Fundamentals of polymer science. `http://nsdl.niscair.res.in/jspui/bitstream/123456789/406/2/Molecular%20weights%20of%20polymers.pdf`. Accessed: 2018-01-04.

[143] Polymer molecular weight distribution and definitions of mw averages. `https://www.agilent.com/cs/library/technicaloverviews/public/5990-7890EN.pdf`. Accessed: 2018-01-04.

[144] J. E. Scott, C. Cummings, A. Brass, and Y. Chen. Secondary and tertiary structures of hyaluronan in aqueous solution, investigated by rotary shadowing-electron microscopy and computer simulation. *Biochemical Journal*, 274:699–705, 1991.

[145] F. Heatley and J. E. Scott. A water molecule participates in the secondary structure of hyaluronan. *Biochemical Journal*, 254:489–493, 1988.

[146] H. S. Azevedo and R. L. Reis. Understanding the enzymatic degradation of biodegradable polymers and strategies to control their degradation rate. In R. L. Reis and J. S. Roman, editors, *Biodegradable Systems in Tissue Engineering and Regenerative Medicine*, chapter 12, pages 177–201. CRC Press, 2005.

[147] Howard M Katzen and Robert T Schimke. Multiple forms of hexokinase in the rat: tissue distribution, age dependency, and properties. *Proceedings of the National Academy of Sciences*, 54(4):1218–1225, 1965.

[148] Vilberto Stocchi, Mauro Magnani, Franco Canestrari, Marina Dacha, and Giorgio Fornaini. Multiple forms of human red blood cell hexokinase. preparation, characterization, and age dependence. *Journal of Biological Chemistry*, 257(5):2357–2364, 1982.

[149] JE Wilson. Hexokinases. *Reviews of physiology, biochemistry and pharmacology*, 126:65, 1995.

[150] Oliver H Lowry and Janet V Passonneau. The relationships between substrates and enzymes of glycolysis in brain. *Journal of Biological Chemistry*, 239(1):31–42, 1964.

[151] S Rapoport. The regulation of glycolysis in mammalian erythrocytes. *Essays Biochem.*, 4:69–103, 1968.

[152] Vladimir B Ritov and David E Kelley. Hexokinase isozyme distribution in human skeletal muscle. *Diabetes*, 50(6):1253–1262, 2001.

[153] Sami Heikkinen, Suvikki Suppola, Mari Malkki, Samir S Deeb, Juhani Jänne, and Markku Laakso. Mouse hexokinase ii gene: structure, cdna, promoter analysis, and expression pattern. *Mammalian Genome*, 11(2):91–96, 2000.

[154] Siby Sebastian, Seby Edassery, and John E Wilson. The human gene for the type iii isozyme of hexokinase: structure, basal promoter, and evolution. *Archives of biochemistry and biophysics*, 395(1):113–120, 2001.

[155] Maria L Cardenas, A Cornish-Bowden, and T Ureta. Evolution and regulatory role of the hexokinases. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, 1401(3):242–264, 1998.

[156] John E Wilson. Isozymes of mammalian hexokinase: structure, subcellular localization and metabolic function. *Journal of Experimental Biology*, 206(12):2049–2057, 2003.

[157] Patrick B Iynedjian. Mammalian glucokinase and its gene. *Biochemical Journal*, 293(Pt 1):1, 1993.

[158] Richard L Printz, Mark A Magnuson, and Daryl K Granner. Mammalian glucokinase. *Annual review of nutrition*, 13(1):463–496, 1993.

[159] Heftsi Azoulay-Zohar, Adrian Israelson, ABU-HAMAD Salah, and Varda Shoshan-Barmatz. In self-defence: hexokinase promotes voltage-dependent anion channel closure and prevents mitochondria-mediated apoptotic cell death. *Biochemical Journal*, 377(2):347–355, 2004.

[160] Andrew P Halestrap, Gavin P McStay, and Samantha J Clarke. The permeability transition pore complex: another view. *Biochimie*, 84(2-3):153–166, 2002.

[161] Ana Preller and John E Wilson. Localization of the type iii isozyme of hexokinase at the nuclear periphery. *Archives of biochemistry and biophysics*, 294(2):482–492, 1992.

[162] Eugene Wyatt, Rongxue Wu, Wael Rabeh, Hee-Won Park, Mohsen Ghanefar, and Hossein Ardehali. Regulation and cytoprotective role of hexokinase iii. *PloS one*, 5(11):e13823, 2010.

[163] PB Iynedjian. Molecular physiology of mammalian glucokinase. *Cellular and Molecular Life Sciences*, 66(1):27, 2009.

[164] Tracy K White and John E Wilson. Isolation and characterization of the discrete n-and c-terminal halves of rat brain hexokinase: retention of full catalytic activity in the isolated c-terminal half. *Archives of Biochemistry and Biophysics*, 274(2):375–393, 1989.

[165] Krishan K Arora, Charles R Filburn, and Peter L Pedersen. Structure/function relationships in hexokinase. site-directed mutational analyses and characterization of overexpressed fragments implicate different functions for the n-and c-terminal halves of the enzyme. *Journal of Biological Chemistry*, 268(24):18259–18266, 1993.

[166] Chenbo Zeng and Herbert J Fromm. Active site residues of human brain hexokinase as studied by site-specific mutagenesis. *Journal of Biological Chemistry*, 270(18):10509–10513, 1995.

[167] Hossein Ardehali, Richard L Printz, Richard R Whitesell, James M May, and Daryl K Granner. Functional interaction between the n-and c-terminal halves of human hexokinase ii. *Journal of Biological Chemistry*, 274(23):15986–15989, 1999.

[168] Hossein Ardehali, Yutaka Yano, Richard L Printz, Steve Koch, Richard R Whitesell, James M May, and Daryl K Granner. Functional organization of mammalian hexokinase ii retention of catalytic and regulatory functions in both the nh-and cooh-terminal halves. *Journal of Biological Chemistry*, 271(4):1849–1852, 1996.

[169] Douglas M Schirch and John E Wilson. Rat brain hexokinase: location of the substrate hexose binding site in a structural domain at the c-terminus of the enzyme. *Archives of biochemistry and biophysics*, 254(2):385–396, 1987.

[170] Henry J Tsai and John E Wilson. Functional organization of mammalian hexokinases: characterization of the rat type iii isozyme and its chimeric forms, constructed with the n-and c-terminal halves of the type i and type ii isozymes. *Archives of biochemistry and biophysics*, 338(2):183–192, 1997.

[171] Henry J Tsai. Functional organization and evolution of mammalian hexokinases: mutations that caused the loss of catalytic activity in n-terminal halves of type i and type iii isozymes. *Archives of biochemistry and biophysics*, 369(1):149–156, 1999.

[172] James Ning, Daniel L Purich, and Herbert J Fromm. Studies on the kinetic mechanism and allosteric nature of bovine brain hexokinase. *Journal of Biological Chemistry*, 244(14):3840–3846, 1969.

[173] Gerhard Gerber, Heidemarie Preissler, Reinhart Heinrich, and Samuel M Rapoport. Hexokinase of human erythrocytes: purification, kinetic model and its application to the conditions in the cell. *European journal of biochemistry*, 45(1):39–52, 1974.

[174] Daniel L Purich and Herbert J Fromm. The kinetics and regulation of rat brain hexokinase. *Journal of Biological Chemistry*, 246(11):3456–3463, 1971.

[175] JE Wilson. Brain hexokinase, the prototype ambiquitous enzyme. In *Current Topics in Cellular Regulation*, volume 16, pages 1–44. Elsevier, 1980.

[176] Frederick B Rudolph and Herbert J Fromm. Computer simulation studies with yeast hexokinase and additional evidence for the random bi bi mechanism. *Journal of Biological Chemistry*, 246(21):6611–6619, 1971.

[177] Camillo Rosano, Elisabetta Sabini, Menico Rizzi, Daniela Deriu, Garib Murshudov, Marzia Bianchi, Giordano Serafini, Mauro Magnani, and Martino Bolognesi. Binding of non-catalytic atp to human hexokinase i highlights the structural components for enzyme–membrane association control. *Structure*, 7(11):1427–1437, 1999.

[178] Xiaofeng Liu, Chang Sup Kim, Feruz T Kurbanov, Richard B Honzatko, and Herbert J Fromm. Dual mechanisms for glucose 6-phosphate inhibition of human brain hexokinase. *Journal of Biological Chemistry*, 274(44):31155–31159, 1999.

[179] Siby Sebastian, John E Wilson, Anne Mulichak, and R Michael Garavito. Allosteric regulation of type i hexokinase: a site-directed mutational study indicating location of the functional glucose 6-phosphate binding site in the n-terminal half of the enzyme. *Archives of biochemistry and biophysics*, 362(2):203–210, 1999.

[180] Tsuei-Yun Fang, Olga Alechina, Alexander E Aleshin, Herbert J Fromm, and Richard B Honzatko. Identification of a phosphate regulatory site and a low affinity binding site for glucose 6-phosphate in the n-terminal half of human brain hexokinase. *Journal of Biological Chemistry*, 273(31):19548–19553, 1998.

[181] PA Lazo, A Sols, and JE Wilson. Brain hexokinase has two spatially discrete sites for binding of glucose-6-phosphate. *Journal of Biological Chemistry*, 255(16):7548–7551, 1980.

[182] Warren Ross Ellison, JD Lueck, and HJ Fromm. Studies on the mechanism of orthophosphate regulation of bovine brain hexokinase. *Journal of Biological Chemistry*, 250(5):1864–1871, 1975.

[183] Kenneth B Taylor. *Enzyme kinetics and mechanisms*, volume 64. Springer Science & Business Media, 2002.

[184] ODEINT. The `odeint` solver in the `integrate` module of the scipy library. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html`, -. Accessed: 2019-4-19.

[185] Gert Rijksen and Gerard EJ Staal. Regulation of human erythrocyte hexokinase: The influence of glycolytic intermediates and inorganic phosphate. *Biochimica et Biophysica Acta (BBA)-Enzymology*, 485(1):75–86, 1977.

[186] Chin-Rang Yang, Bruce E Shapiro, Eric D Mjolsness, and G Wesley Hatfield. An enzyme mechanism language for the mathematical modeling of metabolic pathways. *Bioinformatics*, 21(6):774–780, 2004.

[187] Irwin H Segel. *Enzyme kinetics: behavior and analysis of rapid equilibrium and steady state enzyme systems.* Wiley New York, 1993.

[188] JAMES E Foley, SAMUEL W Cushman, and LESTER B Salans. Intracellular glucose concentration in small and large rat adipose cells. *American Journal of Physiology-Endocrinology And Metabolism*, 238(2):E180–E185, 1980.

[189] John E Wilson. Brain hexokinase a proposed relation between soluble-particulate distribution and activity in vivo. *Journal of Biological Chemistry*, 243(13):3640–3647, 1968.

[190] Ron Milo and Rob Phillips. *Cell biology by the numbers.* Garland Science, 2015.

[191] Scott John, James N Weiss, and Bernard Ribalet. Subcellular localization of hexokinases i and ii directs the metabolic fate of glucose. *PloS one*, 6(3):e17674, 2011.

[192] D Andrew Skaff, Chang Sup Kim, Henry J Tsai, Richard B Honzatko, and Herbert J Fromm. Glucose 6-phosphate release of wild-type and mutant human brain hexokinases from mitochondria. *Journal of Biological Chemistry*, 280(46):38403–38409, 2005.

[193] Irwin A Rose and Jessie VB Warms. Mitochondrial hexokinase release, rebinding, and location. *Journal of Biological Chemistry*, 242(7):1635–1645, 1967.

[194] Brian P Ingalls. *Mathematical modeling in systems biology: an introduction.* MIT press, 2013.

[195] David Fell and Athel Cornish-Bowden. *Understanding the control of metabolism*, volume 2. Portland press London, 1997.

[196] Hubert M Piwonski, Mila Goomanovsky, David Bensimon, Amnon Horovitz, and Gilad Haran. Allosteric inhibition of individual enzyme molecules trapped in lipid vesicles. *Proceedings of the National Academy of Sciences*, 109(22):E1437–E1443, 2012.

[197] Jenni Rahikainen, Saara Mikander, Kaisa Marjamaa, Tarja Tamminen, Angelos Lappas, Liisa Viikari, and Kristiina Kruus. Inhibition of enzymatic hydrolysis by residual lignins from softwood—study of enzyme binding and inactivation on lignin-rich surface. *Biotechnology and bioengineering*, 108(12):2823–2834, 2011.

[198] Christine Bobin-Dubigeon, Marc Lahaye, Fabienne Guillon, Jean-Luc Barry, and Daniel J Gallant. Factors limiting the biodegradation of ulva sp cell-wall polysaccharides. *Journal of the Science of Food and Agriculture*, 75(3):341–351, 1997.

[199] J Bernhard Wehr, Neal W Menzies, and F Pax C Blamey. Inhibition of cell-wall autolysis and pectin degradation by cations. *Plant Physiology and Biochemistry*, 42(6):485–492, 2004.

[200] Uwe Bornscheuer, Klaus Buchholz, and Juergen Seibel. Enzymatic degradation of (ligno) cellulose. *Angewandte Chemie International Edition*, 53(41):10876–10893, 2014.

[201] Reenu Anne Joy, Narendranath Vikkath, and Prasanth S Ariyannur. Metabolism and mechanisms of action of hyaluronan in human biology. *Drug metabolism and personalized therapy*, 33(1):15–32, 2018.

[202] Eva A Turley, David K Wood, and James B McCarthy. Carcinoma cell hyaluronan as a “portable” cancerized prometastatic microenvironment. *Cancer research*, 76(9):2507–2512, 2016.

[203] James B McCarthy, Dorraya El-Ashry, and Eva A Turley. Hyaluronan, cancer-associated fibroblasts and the tumor microenvironment in malignant progression. *Frontiers in cell and developmental biology*, 6:48, 2018.

[204] Xiao Tian, Jorge Azpurua, Christopher Hine, Amita Vaidya, Max Myakishev-Rempel, Julia Ablaeva, Zhiyong Mao, Eviatar Nevo, Vera Gorbunova, and Andrei Seluanov. High-molecular-mass hyaluronan mediates the cancer resistance of the naked mole rat. *Nature*, 499(7458):346, 2013.

[205] Nigel C Veitch. Horseradish peroxidase: a modern view of a classic enzyme. *Phytochemistry*, 65(3):249–259, 2004.

[206] Thermo Fisher Scientific. Amplex^{tm} red reagent. `https://www.thermofisher.com/order/catalog/product/A12222#/A12222`. Accessed: 2019 - 10 - 09.