MODERN MATHEMATICAL METHODS IN MODELING AND DYNAMICS OF REGULATORY SYSTEMS OF GENE-ENVIRONMENT NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZLEM DEFTERLİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MATHEMATICS

AUGUST 2011

Approval of the thesis:

MODERN MATHEMATICAL METHODS IN MODELING AND DYNAMICS OF REGULATORY SYSTEMS OF GENE-ENVIRONMENT NETWORKS

submitted by **ÖZLEM DEFTERLİ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mathematics Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Zafer Nurlu
Head of Department, **Mathematics** _____

Assoc. Prof. Dr. Songül Kaya Merdan
Supervisor, **Department of Mathematics, METU** _____

Prof. Dr. Gerhard-Wilhelm Weber
Co-supervisor, **Institute of Applied Mathematics, METU** _____

Examining Committee Members:

Prof. Dr. Marat Akhmet
Department of Mathematics, METU _____

Assoc. Prof. Dr. Songül Kaya Merdan
Department of Mathematics, METU _____

Assoc. Prof. Dr. Tolga Can
Department of Computer Engineering, METU _____

Prof. Dr. Ömer L. Gebizlioğlu
Department of Statistics, Ankara University _____

Assist. Prof. Dr. Hakan Öktem
Institute of Applied Mathematics, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ÖZLEM DEFTERLİ

Signature :

ABSTRACT

MODERN MATHEMATICAL METHODS IN MODELING AND DYNAMICS OF REGULATORY SYSTEMS OF GENE-ENVIRONMENT NETWORKS

Defterli, Özlem
Ph.D., Department of Mathematics
Supervisor : Assoc. Prof. Dr. Songül Kaya Merdan
Co-Supervisor : Prof. Dr. Gerhard-Wilhelm Weber
August 2011, 150 pages

Inferring and anticipation of genetic networks based on experimental data and environmental measurements is a challenging research problem of mathematical modeling.

In this thesis, we discuss gene-environment network models whose dynamics are represented by a class of time-continuous systems of ordinary differential equations containing unknown parameters to be optimized. Accordingly, time-discrete version of that model class is studied and improved by using different numerical methods. In this aspect, 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method are newly introduced, iteration formulas are derived and corresponding matrix algebras are newly obtained.

We use nonlinear mixed-integer programming for the parameter estimation and present the solution of a constrained and regularized given mixed-integer problem. By using this solution and applying the 3^{rd}-order Heun's and 4^{th}-order classical Runge-Kutta methods in the time-discretized model, we generate corresponding time-series of gene-expressions by this thesis. Two illustrative numerical examples are studied newly with an artificial data set and a real-world data set which expresses a real phenomenon. All the obtained approximate results are compared to see the goodness of the new schemes. Different step-size analysis and sensitivity

tests are also investigated to obtain more accurate and stable predictions of time-series results for a better service in the real-world application areas.

The presented time-continuous and time-discrete dynamical models are identified based on given data, and studied by means of an analytical theory and stability theories of rarefication, regularization and robustification.

Keywords: dynamical systems, gene-environment regulatory networks, discretization methods and comparisons, mixed-integer nonlinear programming, regularization

ÖZ

GEN-ORTAM AĞLARININ DÜZENLEYİCİ SİSTEMLERİNİN DİNAMİKLERİ VE MODELLEMESİNDE MODERN MATEMATİKSEL YÖNTEMLER

Defterli, Özlem
Doktora, Matematik Bölümü
Tez Yöneticisi : Doç. Dr. Songül Kaya Merdan
Ortak Tez Yöneticisi : Prof. Dr. Gerhard-Wilhelm Weber

Ağustos 2011, 150 sayfa

Deneysel verilere ve çevresel ölçümlere dayanarak genetik ağların çıkarımının ve tahmininin yapılması ilgi çekici bir matematiksel modelleme araştırma problemidir.

Bu tezde, dinamiği bir sınıf zaman-sürekli ve optimize edilmesi gereken parametreler içeren adi diferansiyel denklem sistemleri kullanılarak ifade edilmiş gen-ortam ağ modelleri üzerinde tartıştık. Buna göre, bu model sınıfının zaman-ayrık versiyonları çalışılmış ve farklı sayısal yöntemler kullanılarak geliştirilmiştir. Bu yönde, üçüncü dereceden Heun metodu ve dördüncü dereceden klasik Runge-Kutta yöntemi yeni olarak tanıtılmış, iterasyon formülleri türetilmiş ve ilgili matris cebiri yeni elde edilmiştir.

Parametre tahmininde doğrusal olmayan karma-tamsayılı programlama kullandık ve verilen kısıtlı, düzenlenmiş karma-tamsayılı bir problemin çözümünü sunduk. Bu tezle çalışmasıyla, bu çözümü kullanarak ve zaman-ayrıklaştırılmış modelde üçüncü dereceden Heun metodunu ve dördüncü dereceden klasik Runge-Kutta metodlarını uygulayarak, gen-ekspresyonlarının ilgili zaman serilerini oluşturduk. Hem yapay bir veri dizisi ve hem de gerçek-dünyadan gerçek bir fenomeni ifade bir veri dizisi kullanılarak iki aydınlatıcı sayısal örnek bu tezde yeni olarak incelenmiştir. Bu yeni yöntemlerin etkinliğini görmek için de, elde edilen tüm yaklaşık

sonuçlar karşılaştırılmıştır. Gerçek dünyadaki uygulama alanlarında daha iyi hizmet vermek adına zaman-serilerinin tahmin sonuçlarının daha kesin ve kararlı elde edilmesi amacıyla farklı adım boyutları analizleri ve duyarlılık testleri incelenmiştir.

Sunulan zaman-sürekli ve zaman-ayrık modellerin verilen verilere göre tespiti yapılmış, analitik teorisi ve kararlılık teorileri sağlamlaştırma ve düzenleme yönü ile çalışılmıştır.

Anahtar Kelimeler: dinamik sistemler, gen-çevre düzenleyici ağları, ayrıklaştırma yöntemleri ve karşılaştırmaları, doğrusal olmayan karma-tamsayılı programlama, düzenleme

To my parents,
my sister and my brother

ACKNOWLEDGMENTS

I would like to deeply thank to all those people who supported me during my PhD study. Firstly, I am grateful to my supervisor Assoc. Prof. Dr. Songül Kaya Merdan for her help, guidance, useful comments and for her understanding. I would like to express my gratitude to my co-supervisor Prof. Dr. Gerhard-Wilhelm Weber for his encouragement, endless support and for all his efforts in the preparation of this thesis. In this period, I had the opportunity to make valuable collaborations both with national and international scientists. I always appreciate the friendship, patience and kindness of my supervisor and co-supervisor. Thus, I deeply indebted to them for giving me the chance to complete this thesis.

I would also like to take this opportunity to thank to the thesis defense committee members, Prof. Dr. Marat Akhmet, Assoc. Prof. Dr. Tolga Can, Prof. Dr. Ömer Gebizlioğlu and Assist. Prof. Dr. Hakan Öktem for their important remarks, suggestions and corrections which are useful and essential in the improvement of this thesis.

I am very glad to be able to collaborate with Dr. Armin Fügenschuh, Dr. Erik Kropat, Dr. Sırma Z. Alparslan Gök, Ayşe Özmen and Zehra Çavuşoğlu during our common research project and thanks to all for the exchange of ideas. I express my special thanks to Dr. Armin Fügenschuh for his continuous support and great help in providing the necessary software and algorithms for the optimization part of the work. Many thanks to Prof. Dr. Oliver Stein and Dr. Erik Kropat for their careful comments. Additionally, I would like to thank Prof. Dr. Röbbe Wünschiers for helping me to understand some of the biological notions and for his comments on the biological interpretations of our numerical results.

Thanks to all my friends and my colleagues for their motivation to complete this thesis. I acknowledge the important help, valuable discussions and encouragement of my former M.Sc. supervisor Prof. Dr. Dumitru Baleanu. Moreover, I wish to thank Assist. Prof. Dr. Vilda Purutçuoğlu and Dr. Armin Fügenschuh for their important remarks and kind help in proofreading.

I am also thankful to the Scientific and Technical Research Council of Turkey (TUBITAK) for the financial support.

Finally, I would like to thank my family for their continuous support, care and endless love. I appreciate important help and motivation of my dear sister and thank to her. I would like to dedicate this study to my lovely family whom I am always proud of.

TABLE OF CONTENTS

LIST OF SYMBOLS

$\mathbb{Q}$	:	the set of rational numbers
$\mathbb{Z}$	:	the set of all integers
$\mathbb{R}$	:	the set of all real numbers
$\mathbb{N}$	:	the set of all natural numbers
$\mathbb{N}_0$	:	the set of all nonnegative integers
$\mathbb{R}^n$	:	$n-$dimensional real Euclidean space, $n \in \mathbb{N}$
$C^k(A \to B)$	:	the set of functions from A to B having a continuous derivative of order k
$C^k(A)$	:	the set of functions from A to A having a continuous derivative of order k
$\dot{x} = \dot{x}(t) = \frac{dx}{dt}$	:	derivative of x with respect to t
$N_\delta(x)$	:	δ-neighborhood of the point x for $\delta > 0$

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

System biology is one of the most interesting and important scientific fields of the new century. There still exist open problems based on the analysis and reconstruction of biological systems. In order to develop models with simulations for this purpose, a detailed understanding of regulation processes at the molecular level is required. Such a work can be successful with interdisciplinary cooperations between biologists, mathematicians and computer scientists where also experimental techniques are needed [1].

Modern *Deoxyribonucleic acid (DNA)* sequencing methods analyze large DNA sequences within a reasonable time and they serve the fundamental information to determine the potential components of regulatory networks. High-throughput techniques, i.e., DNA-microarray technology, allow to measure concentrations of all gene products of a cell simultaneously. By using the data obtained from these measurement techniques, construction of models for predicting the global behavior of a biological system have a great importance and interest for the future [1].

1.1 Gene-Regulatory Networks and the Modeling Approaches

The development of the high-throughput techniques gives the possibility to go further in biological research and do investigations on a system-level approach rather than single cell components, which aims at an understanding of interactions between these cell components. This objective requires modeling and analysis methods for these regulatory networks [1]. With the cooperation of biochemical and genetics studies and user friendly computer tools, various mathematical models have been constructed to describe gene interactions and to make

predictions in a systematic way [2]. Additionally, the development of new mathematical methods for the analysis of these highly interconnected systems allows a deeper understanding in the dynamic behavior and the topological aspects of complex regulatory systems appearing not only in biology, but also in finance, engineering and environmental sciences. Modeling gene networks and network reconstruction from experimental data can be seen further in the reviews [2, 3, 4, 5, 6, 7, 8].

The modeling approaches can be summarized as:

- Modeling by graphs,
- Bayesian networks,
- Boolean networks,
- Dynamic models derived from ordinary differential equations or piecewise linear differential equations and hybrid system modeling.

Modeling by graphs is a very common way of modeling gene regulatory networks by representing them as directed graphs. A directed graph (or network) is expresses by $G = (V, A)$ consisting a finite set of nodes (vertices), V, and a subset A of arcs of the cross product $V \times V$ [9]. Here, an arc $a \in A$ is defined as an ordered pair of distinct nodes $(v_1, v_2) \in V \times V$ in the directed graph. In a genetic network, the nodes refer to genes and the arcs (edges) refer to the relationships between the genes. The pair $a = (v_1, v_2)$ of nodes is an arc which connects two nodes (genes), v_1 and v_2, and weighted with a value. This value represents the influence of the first node v_1 (gene 1) on the second node v_2 (gene 2). Moreover, this associated weight can be positive or negative which denotes for an activation effect or inhibition effect, respectively, between gene 1 and gene 2 [10].

By looking at the operations on the regulatory networks, we can get information about biological features. These operations can be the pathways between two predefined genes, cycles, connectivity, etc., which give us some hints about missing regulatory relations, connectivity, structures (components and clusters) and about redundancy in the network [11].

Boolean networks are deterministic dynamic models and firstly studied by Kauffman [12] in order to model gene regulation. From that time, these models have commonly been used to describe the dynamic of gene regulatory networks [13, 14, 15, 16, 17, 18]. The advantage

of Boolean networks is that they can present a rich variety of dynamic behaviors such as convergence to a stable steady state, multi-stationarity, oscillations, switch-like behavior or hysteresis [19, 20]. Since Boolean networks are in the class of deterministic models, they cannot capture the stochastic nature of gene expression and do not account for noise in the measurements. Many extensions of Boolean networks have been proposed in order to overcome some of its limitations (for explanations, see [1] and its cited references). Among these extensions, the most important are probabilistic Boolean networks [21] which were developed to be able to consider and include the stochastic fluctuations in the regulation processes [1].

Bayesian networks are firstly used by Murphy and Mian [22] to model gene interactions. These network models are still frequently used to reconstruct interactions among genes by using expression data [23, 24]. Getting information for a Bayesian network from experimental data is related with estimating the joint probability distribution which defines the structure of the corresponding directed acyclic graph [1]. Bayesian networks are stochastic systems. They consider stochastic effects and so account for stochastic fluctuations in regulation processes and noisy measurements. On the other hand, they can not express the evolution over time since they are static models. Also, the underlying interaction graph has to be acyclic in order to obtain a well-defined joint probability distribution. These are the limitations of Bayesian networks to model complex phenomena such as oscillations, multi-stationarity and hysteresis [5, 19, 20]. Dynamic Bayesian networks [22, 24] are developed by various scientists in order to deal with these restrictions [1].

Further information and more references about these modeling techniques can be seen from [1, 2, 11, 25]. The methods listed above have both advantages and disadvantages [2, 26] with respect to the goodness of data fit, computation time, capturing dynamics, stability and other qualitative or quantitative aspects.

Modeling with ordinary differential equations is developed in recent years, to represent the dynamic behavior of gene regulatory networks quantitatively. This type of models leads to a better understanding of underlying mechanisms causing certain kinds of dynamic behaviors in comparison to previously mentioned Boolean and Bayesian networks. Different parametrizations of the right-hand side function of the initial value problem have been suggested. The parameters in the model refer directly to reaction rates, binding affinities or degradation rates, which are useful both for a reasonable restriction of the parameter space and the interpretation

of inference results [1].

A linear ordinary differential equation model is firstly proposed by Chen et al. [27] for the reconstruction of gene interactions from measured gene expression data. Further extensions of ordinary differential equation models include piecewise linear and nonlinear approaches, hybrid system approach (see [11, 25, 28, 29, 30, 31, 32, 33] and references therein), stochastic kinetic approaches, partial differential equations and delay differential equations [34]. In [28], the mixed continuous-discrete model is introduced to contain the most relevant regulating interactions in a cell as a complementary approach to the one firstly given by [27] and later on by [11, 25, 35]. In this model, a complete description of the dynamics is given by a hybrid system.

Various mathematical methods which have been developed for the construction and analysis of such networks can be seen in [6, 9, 10, 11, 25, 27, 28, 36, 37, 38, 39, 40, 41, 42] and related citations. In [11, 25, 29, 31, 32, 42], a more sophisticated model of differential equations is studied by the inclusion of an additive shift term which leads into an extended space of model functions. *Gene-environment networks* and embedding of the genetic networks into them are newly introduced in [32, 39, 40]. Here, the new nodes are environmental items such as, e.g., toxins, radiation in terms of biology. Moreover, in the application areas of gene-environment networks, the possible environmental factors can be poison in soil, groundwater, air and food, global warming, different types of radiation and electro-magnetic waves. Furthermore, welfare and general items of lifestyle in a society, but also education and campaigns for a more healthy lifestyle can also be regarded as environmental items. Since advanced high technology methods, like DNA microarray technology, are very valuable but also affected with various uncertainties, noise and measurement errors, then those errors are also included in gene-environment network models by considering different kinds of uncertainties. Those uncertainty and error sets are in the types of interval, polyhedral and ellipsoidal uncertainty sets where bounds on the uncertain variable are imposed (see for example [29, 30, 43, 44, 45, 46, 47] and cited references therein). Finally, the robust versions of considered gene-environment network models are newly studied by the application of most modern methods in [48, 49].

In this thesis, we consider the dynamical modeling approach for the representation of the system dynamics of regulatory networks where it is represented by the system of ordinary

differential equations.

1.2 Aspects of Rarefication, Regularization on the Level of Modeling these Networks

Mathematically speaking, in the solution of inverse problems, sometimes we can face with ill-posed problems in the case of continuous systems, or ill-conditioned in the case of discrete linear systems. It is usually possible to stabilize the inversion process by imposing additional constraints that bias the solution. Such a process is generally called as *regularization*. which is a common method to deal with an ill-posed or ill-conditioned inverse problem [50].

A problem is defined as *ill-posed* problem if a solution is not existing or not unique or if it is not stable under perturbation on data. *Tikhonov regularization* is the most common and well-known form to make these problems regular and stable [51].

In the usual descriptions of modeling of gene regulatory networks by systems, the variables denote concentrations of gene products, that are *messenger ribonucleic acids (mRNA)s* or proteins, and the data set contains microarray gene expression measurements. The inference of gene regulatory networks can be formulated as an optimization problem with a given space of state variables, a set of conditions, model functions and observations.

The aim of the network inference problem is to select a model which fits better with the given data which is generally sparse. This sparsity means that the number of network components is large whereas the number of different conditions, or time points, is small at the same time. Hence, there is a fitting problem of a high-dimensional function to only a few data points. Therefore, the corresponding optimization problems are ill-posed. Various regularization methods have been developed to overcome this problem. Some of the well-known regularization methods for stochastic models are the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) [10, 28, 36, 37, 55] which include the negative logarithm of the likelihood function and penalized term having a large number of parameters. The motivated approaches restrict the parameter space by considering biological knowledge in the optimization process. This can be done by introducing constraints into the optimization problem such as upper bounds for single parameters. As another way, penalty terms can be added to the likelihood function [1].

Therefore, these models have to be constructed by taking into account the specific biological knowledge in order to have more reliable predictions [1].

1.3 Scope of the Thesis

In this thesis, the *aim* is to contribute to a further development of mathematical modeling, dynamical systems and optimization theory in the fields of computational biology, engineering and environmental sciences, which are among the most challenging and emerging research areas. In the modeling, prediction and optimization of target-environment and gene-environment regulatory networks, which appear in the mentioned real-world areas, we consider the dynamical modeling approach. We start with time-continuous models to express the evolvement in time of various expression levels of target variables, including their interactions with the environmental variables in the network, and then turn to the time-discrete case. Different and most modern numerical discretization schemes are studied, applied and compared in order to obtain more accurately generated time-series predictions of expression levels of targets. The step-size analysis and sensitivity tests are analyzed. By this thesis work, we give a contribution to the methodology in mathematics in the aspect of improved modeling of target-environment and gene-environment regulatory networks, including their rarefication which may be regarded as a regularization, and to the numerical solution of their dynamics. We present many new ideas and approaches both in the theoretical and applied study of gene-environment networks under mathematical and interdisciplinary considerations. This thesis study gives a mathematical contribution that joins the given toolboxes and methodologies of statistics and data mining.

The new results obtained in this thesis require further improvements of the algorithms and different kinds of rarefication and combined methods, which have to be computationally validated, together with the comparative studies. The studied real-world example can be considered as a first-step implementation of our newly derived explicit numerical methods on real-world data, again with a comparative study. We will combine our new numerical methods with the concepts of correlation, uncertainty and robustness to make modeling and prediction both more accurate and more stable in order to give a better service in the real-world application areas.

These studies mean innovations in the biological and environmental subjects under consideration, and lead to advances in mathematical and applied sciences. Indeed, the areas of applications go much further than gene-environment networks, but enter various other areas, such as finance, which we also regard as an area of the environment with respect to biological and health states. The thesis contributes to the mathematical sciences, while also supporting medical and living conditions of the people.

During the study of this thesis, I have learned and actively applied how to model the time evolution of gene-environment networks (e.g., linear, nonlinear) both in time-continuous and time-discrete cases. Moreover, the original contributions are listed below:

- Time-discretization of the dynamics of the studied model class is improved with most modern numerical schemes. In this respect, 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method are newly introduced and implemented to time-discrete version of the dynamic model class of gene-environment networks,
- Numerical applications and comparisons of these newly introduced methods are firstly studied both with artificial data and real-world data,
- Sensitivity tests and step-size analysis of the time discrete models via numerical simulation are newly investigated in the field,
- Numerical optimization in the solution of the dynamics of that models are done, mainly by nonlinear mixed-integer programming with employing most modern codes,
- As a new approach, process version of generalized partial linear modeling is introduced for gene-environment networks,
- Detection and introduction of a promising research direction for the future is done. We explored and briefly demonstrated future research potentials in the use of algorithmic stability theory, and in the theories of classification and regression under various forms of uncertainty.

1.4 Outline of the Thesis

This thesis is comprised of five main chapters. Mainly, the contents are organized as follows:

Chapter 1 introduces the brief history about the modeling of gene-environment networks as regulatory systems. Then, the notion of regularization of these network models is given. Additionally, the objectives and plan of the study are listed.

Chapter 2 includes the preliminaries part of the thesis. The fundamental biological knowledge is given about gene expressions, regulation mechanisms of gene expression, protein production, DNA microarray analysis and effects of environmental factors. Some basic notions of regression is presented together with a brief introduction about regression methods and some regression models which are needed for the parameter estimation problem. The necessary optimization tools are also mentioned.

In Chapter 3, the dynamical modeling approach is presented for the time evolution of regulatory systems, e.g., gene-environment networks, target-environment networks. The model class which is based on a system of ordinary differential equations is presented with the fundamental descriptions. The time-continuous and time-discrete versions of the models are also listed.

This thesis mainly emphasizes and originally contributes on the numerical methods used for the time discretization of the dynamics of gene-environment regulatory systems for a better and more stable long-time predictions of the gene expression values. In Chapter 4, two new numerical methods are studied for this purpose. Their formalizations and corresponding matrix algebras are newly derived based on the most general form of the model class for gene-environment networks. Moreover, the performance and comparison of these two newly implemented numerical methods are examined by the study of two illustrative examples on a given set of data. The nonlinear mixed-integer least-squares problem is formulated and solved as an optimization problem for the parameter identification of the considered gene-environment network model. In one of the examples, an artificial data set is used containing four different types of behavior. Different step-size analysis and sensitivity tests of these numerical methods are performed on the artificial data and obtained results are presented with the help of figures. In the second numerical example, a real-world data set is used coming from real biological phenomena. Before applying the new numerical methods on this real-

data set, a data analysis is conducted in such a way that a small subset from the whole data set is selected and corresponding biological properties are analyzed based on some biological and mathematical techniques. These properties and information representing the behavior of the considered system are used for the construction of the constraints in the corresponding non-linear mixed-integer optimization problem. A relaxed version and so a more general version of this optimization problem is also studied by extending the constraints. Then, the selected subset of the real-world data is used for the comparison of the new numerical methods. The discussion parts includes the comments made on the obtained numerical results which are unbounded and with a weak fitting property. So, this real-world example requires further considerations and improvements in the continuation of the computational study. In both of these examples, the considered data sets belong to gene-networks without environmental factors and a linear model is studied to represent these systems.

Chapter 5 is organized as an introduction to the study of robustification of regulatory systems. In fact, Generalized Partial Linear modeling is presented and newly developed for the gene-environment networks and target-environment networks. The robustified version of this approach is even announced for the inclusion of the uncertainty concept. As a future work, a new research agenda is presented via Generalized Partial Linear Models for the robust identification of regulatory networks including the computational validation on a given set of data.

Finally, conclusions and an outlook to some further studies are stated in the last chapter.

CHAPTER 2

PRELIMINARIES

In this chapter, some basic notions and tools which are needed throughout the thesis will be briefly introduced.

2.1 Basics of Natural and Environmental Sciences

2.1.1 Gene Expressions, Regulation Processes and DNA-Microarrays

Proteins are fundamental elements in the organization of different processes within a cell. Long folded chains of amino acids construct proteins as macromolecules. There are 20 amino acids which the life is composed of. Proteins carry out many functions which are crucial and fundamental for the survival of the cell. DNA encodes the entirety of proteins that a cell can produce. DNA is a sequence of four kinds of nucleotides which are adenine (A), guanine (G), cytosine (C) and thymine (T). The parts of this sequence that encode proteins are called *genes*. The order of nucleotides in a gene carries the needed information for producing a functional protein [1].

Gene expression is the process of protein synthesis which happens in two steps. The first step is called *transcription* where the nucleotide sequence of a gene is transcribed into an intermediate product called *messenger RNA (mRNA)*. The second step in protein production is *translation* in which mRNA carries the genetic information from the chromosomes to the ribosomes, a cell structure containing *ribosomal RNA (rRNA)* and protein base pairs. So, mRNA serves as a template in this process. Shortly, the transfer of genetic information from

DNA to RNA is called transcription and the passing of information from RNA to proteins called translation. The route of this flow of information is called *central dogma of biology* which is depicted in Figure 2.1 (see [1, 11] and their references).

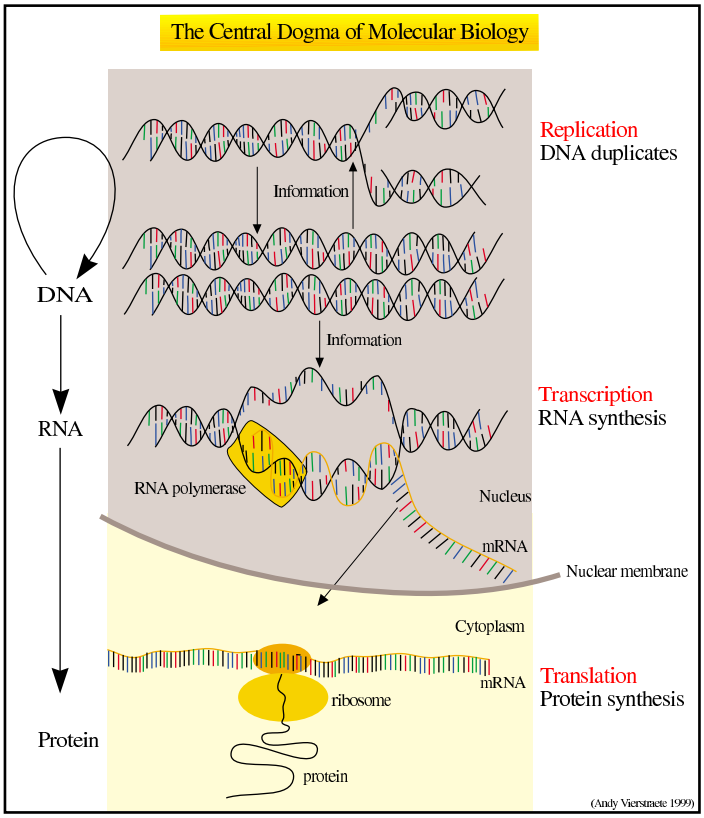


Figure 2.1: Central dogma of biology [52]

The measure of gene expression can be determined from the genomic analysis at the mRNA level [53]. Genomic and environmental factors influence the gene expression levels. For example, the environmental factors including stress, light, temperature and other signals cause some changes in hormones and in enzymatic reactions that affect the gene expression level. Therefore, mRNA analysis tells us information about the genetics of an organism and also about the dynamic changes in environment of that organism. The experimentalists usually measure mRNA levels to determine the future predictions [1, 11].

The rate of gene expression is highly regulated at different levels and can change in a wide

range. This gives the flexibility of adaptation to external conditions, such as nutrition supply, salinity, temperature, and also ability to respond to the perturbations for survival.

A *gene expression pattern* is defined as a snapshot of gene product (mRNA or protein) concentrations. The expression pattern of a cell is determined by the tissue and can be influenced by external conditions. A typical mRNA expression pattern for an organism or a tissue is called *transcriptome* while a protein concentration pattern is called *proteome* [1]. Transcriptional regulation usually happens at the transcription initiation and it affects directly the concentrations of mRNAs, which are measured in microarray experiments. *Transcription factors* are defined as the proteins that bind to the DNA and influence transcription. There has been a huge work to understand binding mechanisms of transcription factors. The models which represent regulation of transcription initiation by binding of proteins to the DNA are referred to as *gene regulatory networks* [1].

In usual methods of molecular biology, the expression level of one gene is obtained in one experiment which results very limited throughput and the whole picture of gene expression is hard to obtain. On the other hand, a new method is developed recently that is called DNA microarray technology [53, 54]. By this technique, the expression of thousands of genes can be monitored in one experiment which also leads to obtain a detailed picture of the interactions between thousands of genes simultaneously [55].

As one of the most interesting problems of molecular and cellular biology, the researchers aim to infer the underlying gene regulatory networks at the system level. This mathematically means a graph consisting of vertices as genes and of edges connecting the genes. Various techniques exist to define a genetic network. In general, a genetic network is described as a collection of molecular components such as a group of genes in which the individual gene can influence or change the activity of other genes [56, 57]. A cellular function is then carried out collectively by the interaction among these genes [10].

A lot of effort has been performed to infer gene regulatory networks and different mathematical methods have been developed for modeling a genetic network mathematically such as differential equation models [27, 36, 58, 59], linear models [60], stochastic models [2, 61], neural networks [62], Bayesian networks [63], and Boolean networks [6].

2.1.2 Effects of the Environment

Interaction between genes and environment is frequently characterized as epigenetic, which refers to stable changes of gene expression patterns in response to environmental factors without any mutations in the DNA sequence. *DNA methylation*, *acetylation*, *ethylation* and *phosphorylation* are some of epigenetic factors providing important *epigenetic regulations* [32, 64, 65]. The fact that the frequency of an epigenetic effect is higher than genetic sequence mutations gives importance to the studies on epigenetics. Therefore, to bring a better explanation of the complexity of nature, the genetic networks cannot be studied alone without taking into consideration the environmental factors which affect epigenetic patterns and, thus, gene expression patterns [42].

Nowadays, it is clearly understood that environmental factors form an important group of regulating components and by including these additional variables the models' performance can be significantly improved [44]. Such a consideration gives advantage as it is presented in [66], where it is shown that prediction and classification performances of supervised learning methods for the most complex genome-wide human disease classification can be greatly improved by considering environmental aspects. Various examples from biology and life sciences, e.g., metabolic networks [43, 67, 68], immunological networks [69], social and ecological networks [70], refers to target-environment and gene-environment regulatory systems where environmental effects are strongly involved (see [44] and its cited references for details).

2.2 Parameter Estimation

For the inference of the regulatory networks, which appears in system biology, environmental sciences, education and economy, it is important to identify, predict and model the relationship among the components (variables).

Regression is one of the common approach used in the literature for this purpose. There exist various methods to handle regression problems but the mostly preferred ones are the *Maximum Likelihood Estimation (MLE)* and *Least-Squares Estimation (LSE)*. These methods are quite newly used for the parameter identification of the gene-environment networks and target-environment networks, as regulatory systems, in many studies with different approaches

and extensions (see [11, 28, 29, 30, 31, 32, 36, 37, 45] and the cited references inside).

2.2.1 Regression

As one of the mathematical and statistical methods, regression analysis is very useful for many types of problems appearing in the areas of engineering and science. Mainly, it analyzes and tries to model the relationship between the dependent variable and one or more independent variables. Regression analysis is widely used for both prediction and estimation, and most commonly estimates the conditional expectation of the dependent variable given the independent variables. The target of the estimation is the regression function, which is called the function of independent variables [71, 72].

There are many regression models in the literature, with applications from different fields (see [74, 79, 80, 81, 82] and references therein), like:

- Linear regression models,
- Nonlinear regression models,
- Generalized linear models,
- Nonparametric regression models,
- Additive models,
- Generalized additive models.

2.2.1.1 Linear Regression

As a statistical method, linear regression correlates the amount of change in the dependent (response or target) variable to the independent (regressor or predictor) variable(s). Here, the model is not necessarily linear in the independent variables but it depends linearly on the unknown parameters and has a linearly additive relationship. In general, the form of a *Linear Regression Model (LRM)* [71, 72, 75, 80] with k regressor variables $x_1, x_2, \ldots, x_k$ is given as follows:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon, \tag{2.1}$$

where ε is a random error with zero mean and unknown variance σ^2, Y is the response variable and x_i ($i = 1, 2, \ldots, k$) represent the independent variables. Additionally, the random errors corresponding to different observations are uncorrelated random variables and assumed to be normally distributed. The conditional expected value of Y for each value of x_i, i.e., by the vector $\boldsymbol{X}$, is given as

$$E(Y \mid \boldsymbol{X}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k. \tag{2.2}$$

We can clearly define the N observations in the sample as

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \quad (i = 1, 2, \ldots, N), \tag{2.3}$$

where N is the number of the data, the errors ε_i are assumed to be uncorrelated random variables with zero mean and variance σ^2 [73, 80].

In order to estimate unknown regression parameters in the above regression problems, one can use MLE and LSE methods where the latter one is the easiest and most common. If the distribution of the errors is known, then MLE is an alternative estimation method which is more general and in some cases more efficient. With both methods the aim is to obtain the line which best predicts the response variable from a given set of data [50, 72, 76].

One can use the LSE method to find the unknown parameters of the general linear regression problem given in (2.3) by minimizing the function of the residual sum of the squares (RSS) between y_i and its expected values:

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{k} x_{ij}\beta_j)^2, \tag{2.4}$$

Here, $RSS(\boldsymbol{\beta})$ is a quadratic function of the parameters. The matrix form of Eq. (2.4) can be written as [76]:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.5}$$

where $\mathbf{y}$ is the $(N \times 1)$-vector of dependent variables, $\boldsymbol{\varepsilon}$ is the $(N \times 1)$ random error vector, $\boldsymbol{\beta}$ is the $(k + 1) \times 1$-vector of unknown parameters, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$, and $\mathbf{X}$ is the $N \times (k + 1)$ (design) matrix of the independent variable and defined by the input data $x_{i,j}$ ($i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, k$):

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} \\ \vdots & \vdots & & \vdots \\ 1 & x_{N1} & \cdots & x_{Nk} \end{pmatrix}. \tag{2.6}$$

We note that (2.4) can be rewritten in terms of the Euclidean norm, $\| \cdot \|_2$, in the following way [76]:

$$RSS(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \|_2^2 . \tag{2.7}$$

Then, corresponding normal equations can be obtained and solved accordingly (see [50, 76] for more details).

LRM is a regression model which is commonly used. A first extension of the linear modeling which permits the models to fit with the data having probability distributions other than the normal distribution are the *generalized linear models*. Most of the important and useful statistical models such a with Poisson, binomial, Gamma or normal distributions can be represented as generalized linear models by selecting an appropriate link function and a response probability distribution. The ordinary linear models are recovered as a special case when the identity function is chosen as the link along with the normal distribution [74, 77, 80, 81].

Both linear models and generalized linear models are influenced by multi-collinearity, missing variables and outliers in the given data. It is not easy to process generalized linear models for choosing important predictors and their interactions [80, 81]. All these difficulties can be managed by using *data mining* which is an interdisciplinary and difficult scientific research works on the outcomes of the experiments and all other kinds of measurements. By using the tools of data mining, high-level categorical predictors are handled efficiently. As an adaptive procedure, *Multiple Adaptive Regression Spline (MARS)* is one of the important data mining tool which is very useful and effective for high-dimensional problems. It does not force to have any specific relation among the predictor and dependent variables. However, it is able to estimate the contributions of basis functions and so both the additive and interaction effects of the independent variables can identify the dependent variable. Using MARS together with generalized linear models makes the model-building process relatively faster and more efficient (see [79, 80, 81, 82] and the related references therein).

(i) Generalized Linear Models

In various areas of prediction, regression and even classification problems *Generalized Linear Models (GLMs)* are applicable. GLM approach is used in the case the normality and constant variance assumptions are not satisfied [75]. It has an advantage like the flexibility in addressing a variety of statistical problems and also in the case of the availability of many software

packages.

GLM allows the mean value of a dependent variable, to depend on a linear predictor through a nonlinear link function and allows the probability distribution of the response variable Y, to be any member of an exponential family of distributions. The fundamental formulation of GLM is as follows (see [74, 77, 80, 81] and their references):

$$\eta_i = H(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta} \quad (i = 1, 2, \ldots, N), \tag{2.8}$$

where H is a smooth monotonic *link function*, $\mu_i = E(Y_i)$ expected value of the response variables Y_i, $\mathbf{x}_i$ is the vector of observed value of explanatory variable for the i^{th}-case, $\boldsymbol{\beta}$ is the vector of unknown parameters, and N is the number of data.

After having introduced GLM next, we will go one step further by the class of *generalized partial linear models*.

(ii) Generalized Partial Linear Models

Since GLM is a linear approach like linear modeling, it can not be used for the representation of any system that contains linear and nonlinear items together. Therefore, *Generalized Partial Linear Models (GPLMs)* are developed having an important advantage that consists in some grouping which could be done for the input dimensions or features in order to assign appropriate submodels specifically. This kind of particular representation of submodels gives more accurate and stable (regular) results in the existence of noise in the data (see [74] and the references mentioned there).

A particular type of semiparametric models are the GPLMs, which are the extended version of the GLMs obtained by adding a single nonparametric component to the usual parametric terms. The GPLM is defined by:

$$E(Y \mid \boldsymbol{X}, \boldsymbol{T}) = G(\boldsymbol{X}^T \boldsymbol{\beta} + \varsigma(\boldsymbol{T})), \tag{2.9}$$

with $\varsigma(\cdot)$ as a smooth function to be estimated, $G := H^{-1}$ is a known link function which links the mean of the response variable, $\mu = E(Y \mid \boldsymbol{X}, \boldsymbol{T})$, to the predictors. Moreover, $\boldsymbol{\beta}$ is a $(m \times 1)$-vector of unknown parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^T$, $\boldsymbol{X}$ is an $(m \times 1)$-random vector representing (typically discrete) covariates and $\boldsymbol{T}$ is an $(q \times 1)$-random vector of continuous covariates where both $\boldsymbol{X}$ and $\boldsymbol{T}$ comes from a decomposition of explanatory variables [74, 78, 80, 81].

Then, the general version of the model can be written as follows [74, 78, 80, 81, 82]:

$$H(\mu) = \eta(\boldsymbol{X}, \boldsymbol{T}) = \boldsymbol{X}^T\boldsymbol{\beta} + \varsigma(\boldsymbol{T}) = \sum_{j=1}^{m} X_j\beta_j + \varsigma(\boldsymbol{T}), \tag{2.10}$$

with observation values y_i, $\mathbf{x}_i$ and $\mathbf{t}_i$ $(i = 1, 2, \ldots, N)$,

$$\mu_i = G(\eta_i), \quad \eta_i = H(\mu_i) = \mathbf{x}_i^T\boldsymbol{\beta} + \varsigma(\mathbf{t}_i). \tag{2.11}$$

Various methods exist for the estimation of GPLMs (see [74, 79, 80, 81, 82] and the references inside).

2.2.1.2 Nonlinear Regression

The systems which has nonlinear interactions among the components can not be represented efficiently and fitted with the data by LRMs. Those systems commonly appear in real-life situations and can be estimated by nonlinear regression [83].

In linear regression models, the function relating the mean of response variables to the independent variables is linear in the parameters. When the model contains at least one nonlinear parameter, then it is called as a *nonlinear model* which means that there exists at least one derivative with respect to a parameter must include that parameter. In nonlinear regression models, the function relating the mean of response variables to the independent variables is nonlinear in its parameters [73].

The general form of nonlinear regression models is given as [75]:

$$Y = f(\mathbf{x}, \boldsymbol{\gamma}) + \varepsilon, \tag{2.12}$$

in which $f(\mathbf{x}, \boldsymbol{\gamma})$ represents the expectation function for the nonlinear regression model, $\boldsymbol{\gamma}$ is a $(k \times 1)$-vector of unknown parameters $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_k)^T$, $\mathbf{x}$ is a $(k \times 1)$-vector of regressor variables $\mathbf{x} = (x_1, x_2, \ldots, x_k)^T$ and ε is an uncorrelated random error with zero mean and variances σ_i^2 $(i = 1, 2, \ldots, N)$. The same model with N observations can be expressed in terms of vector notation as the system below [80]

$$\mathbf{y} = \boldsymbol{\varphi}(\boldsymbol{\gamma}) + \boldsymbol{\varepsilon}, \tag{2.13}$$

where $\boldsymbol{\varphi}(\boldsymbol{\gamma}) := (f(\mathbf{x}_1, \boldsymbol{\gamma}), f(\mathbf{x}_2, \boldsymbol{\gamma}), \ldots, f(\mathbf{x}_N, \boldsymbol{\gamma}))^T$ with the vector of residual $\boldsymbol{\varepsilon}$. In literature, different methods exist for nonlinear regression models, e.g., Nonlinear Regression methods, Maximum Likelihood Estimation method, the Gauss-Newton method and the Levenberg-Marquardt Method [82].

2.3 Some Mathematical Programming Tools

In the modeling and optimization of target-environment (gene-environment) regulatory systems, the following optimization tools are some of the methods which are recently used by the researchers [28, 29, 30, 31, 32, 36, 37, 40, 42, 43, 45, 46, 49, 65, 84, 85] to obtain improved results. Among these tools, the conic quadratic programming with GPLM approach will be studied in the continuation of this thesis as a new and promising idea. The starting point of this new idea is presented in Chapter 5, Section 5.2, for robustification purposes.

2.3.1 Conic Quadratic Programming

In [86], it is mentioned that there can be always a way to pass from an optimization problem to an equivalent one with a linear objective. Then, the corresponding conic quadratic representation of the original problem can be written if it contains one of the following functions and sets: a constant function, an affine function, the fractional-quadratic function, hyperbola, the Euclidean norm, the squared Euclidean norm. Hence, the linear least-squares problem can be converted to conic quadratic form when the Euclidean norm is used and the constraints of the optimization problem is defined inside a second-order cone. Then, these type of problems can be solved by *conic quadratic programming (CQP).*

Definition 2.3.1 *[86] An m-dimensional second-order (or Lorentz or ice-cream) cone L^m is defined as*

$$L^m = \{\boldsymbol{x} = (x_1, \ldots, x_m)^T \in \mathbb{R}^m \mid x_m \geq \sqrt{x_1^2 + \ldots + x_{m-1}^2}\}, \quad m \geq 2. \tag{2.14}$$

Definition 2.3.2 *[86] A conic quadratic problem is a conic problem*

$$\begin{aligned} &\min_{\boldsymbol{x}} \quad \boldsymbol{c}^T\boldsymbol{x}, \\ &\textit{subject to (s.t.)} \quad \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \geq_K \boldsymbol{0}, \end{aligned} \tag{2.15}$$

where $\boldsymbol{x}$ is the design vector, $\boldsymbol{c}$ is a given vector of coefficients of the objective function $\boldsymbol{c}^T\boldsymbol{x}$, $\boldsymbol{A}$ is the given constraint matrix, $\boldsymbol{b}$ is the given right hand side vector of the constraints and the cone K is a direct product of many second-order cones, $K = L^{m_1} \times L^{m_2} \times \cdots \times L^{m_r}$. Here, $\geq_K$ stands for $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \in K$.

A CQP problem is an optimization problem with linear objective function and finitely many *ice-cream constraints*

$$\mathbf{A}_i\mathbf{x} - \mathbf{b}_i \geq_{L^{m_i}} \mathbf{0} \quad (i = 1, 2, \ldots, r).$$

Hence, a CQP can be rewritten as

$$\begin{aligned} &\min_{\mathbf{x}} \quad \mathbf{c}^T\mathbf{x}, \\ &\text{s.t.} \quad \mathbf{A}_i\mathbf{x} - \mathbf{b}_i \geq_{L^{m_i}} \mathbf{0} \quad (i = 1, 2, \ldots, r). \end{aligned} \tag{2.16}$$

Note that, linear programming, conic quadratic programming and semi-definite programming are all particular cases of conic programming.

2.3.2 Generalized Semi-Infinite Programming

As one of the optimization problems class, semi-infinite programming (SIP) problems consist of infinitely many constraints and finitely many variables. During the last thirty years, SIP has been studied and improved in many directions through its theory and numerical methods [87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 106]. In recent years, generalized semi-infinite programming (GSIP) is developed and important properties and applications are analyzed by the researchers [92, 97, 100, 101, 102, 103, 105, 106] (see also their cited references).

Definition 2.3.3 *[102, 103] A GSIP is an optimization problem having the form*

$$\begin{aligned} GSIP: \quad & \min \quad f(\boldsymbol{x}) \quad s.t. \quad \boldsymbol{x} \in M, \\ & with \quad M = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid g(\boldsymbol{x}, \boldsymbol{y}) \leq 0 \, for \, all \, \boldsymbol{y} \in Y(\boldsymbol{x}) \,\}, \\ & and \quad Y(\boldsymbol{x}) = \{\, \boldsymbol{y} \in \mathbb{R}^m \mid v_j(\boldsymbol{x}, \boldsymbol{y}) \leq 0, \, for \, all \, j \in Q \,\}. \end{aligned} \tag{2.17}$$

All defining functions $f,\ g,\ v_j$ $(j \in Q = \{1, \ldots, q\})$, are assumed to be real-valued and at least twice continuously differentiable on their respective domains. Moreover, we assume that the set-valued mapping $Y : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is locally bounded, that is, for each $\bar{\mathbf{x}} \in \mathbb{R}^n$ there exists a neighborhood U of $\bar{\mathbf{x}}$ such that $\bigcup_{\boldsymbol{x} \in U} Y(\boldsymbol{x})$ is bounded in $\mathbb{R}^m$.

Here, the possibly infinite index set $Y(\mathbf{x})$ of the semi-infinite inequality constraint is allowed to vary with $\mathbf{x}$ in a GSIP. On the contrary, in a standard semi-infinite optimization problem the index set is fixed, that is, we have $Y(\mathbf{x}) \equiv Y$, and if Y is described by functional constraints, then the vector function v does not depend on $\mathbf{x}$ [102, 103].

Note that, SIP was from early stages of its development related with Chebyshev approximation [87, 88, 99]. On the other hand, reverse Chebyshev approximation can be modelled by GSIP [97, 104]. Both approximations are used in some of the recent works [29, 30, 31, 32, 40, 42, 43, 45, 46] for a better modeling and anticipation of biological, environmental, economical complex systems when the given experimental data contains error or uncertainty.

2.3.3 Nonlinear Mixed-Integer Programming

The study and solution of linear integer programs is one of the central and important topic of discrete optimization. Linear integer programming has an increasing usage in the modeling of many problems that arise in the fields of science, technology, society and business. There are different methodologies for the solutions of integer programming problems, see [107, 108, 109, 110] and their references.

Definition 2.3.4 *[110] An integer program or more general a mixed integer program (MIP) is defined in the form*

$$\begin{aligned} z_{MIP} = \quad & \min \quad \boldsymbol{c}^T\boldsymbol{x}, \\ & s.t. \quad \boldsymbol{A}\boldsymbol{x} \begin{Bmatrix} \leq \\ = \end{Bmatrix} \boldsymbol{b}, \\ & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}, \\ & \boldsymbol{x} \in \mathbb{Z}^N \times \mathbb{R}^C, \end{aligned} \tag{2.18}$$

where $\boldsymbol{A} \in \mathbb{Q}^{M\times(N\cup C)}$, $\boldsymbol{c} \in \mathbb{Q}^{N\cup C}$, $\boldsymbol{b} \in \mathbb{Q}^M$. *The sets* M, N *and* C *are non-empty and finite sets with* N *and* C *are disjoint. Here,* N *and* C *have the following elements as numbers, i.e.,* $N = \{1, \ldots, p\}$ *and* $C = \{p+1, \ldots, n\}$, *without loss of generality.* $\boldsymbol{l} \in (\mathbb{Q} \cup \{-\infty\})^{N\cup C}, \boldsymbol{u} \in (\mathbb{Q} \cup \{\infty\})^{N\cup C}$ *are the vectors which are called lower and upper bounds on* $\boldsymbol{x}$, *respectively. A variable* x_j, $j \in N \cup C$, *is unbounded from below (above), if* $l_j = -\infty$ ($u_j = \infty$). *An integer variable* $x_j \in \mathbb{Z}$ *with* $l_j = 0$ *and* $u_j = 1$ *is called binary.*

Different notions can also be used for (2.18) as follows:

linear program or *LP*, if $N = \emptyset$,

integer program or *IP*, if $C = \emptyset$,

binary mixed integer program, $0-1$ *mixed integer program* or *BMIP*, if all variables x_j ($j \in N$) are binary,

binary integer program, $0-1$ *integer program* or *BIP*, if (2.18) is a BMIP with $C = \emptyset$.

In reality, the formulation of the same problem appears in applications may not be unique. So, it is important to find the appropriate formulation. Sometimes we may not even have the problem itself but just get the problem formulation as given in (2.18). Hence, the appropriate information, to find the solution, have to be taken from the constraint matrix $\mathbf{A}$, the right-hand side vector $\mathbf{b}$ and the objective function $\mathbf{c}$, which is called a *preprocessing phase* of mixed integer programming solvers. Then, the problem is still in the type of (2.18), but containing more information about the inherit structure of the problem. After that, preprocessing also tries to find and eliminate unnecessary information from a MIP solver's point of view [110].

Mixed integer programming problems are in the category of $\mathcal{NP}$-hard (non-deterministic polynomial-time hard) problems [111] with respect to their complexity. The mostly preferred way for solving optimally an $\mathcal{NP}$-hard problem like (2.18) is to attack it from two sides. As a first step, the dual part of the problem has to be considered and a lower bound on the objective function has to be obtained by relaxation. To eliminate some parts of the problem like constraints and/or variables, which make it more difficult and complex, is the central point of relaxation methods. Various methods exist and can be selected according to the part which will be eliminated and how it will be reintroduced. Relaxing the integrality constraints is a mainly used technique to obtain a linear program and reintroduce the integrality by adding cutting planes [110].

There are different methods to solve MIPs where semi-definite programming is also included. One should notice that, semi-definite programming can be interpreted as a special case of standard semi-infinite programming [112, 113]. Moreover, as a modern and recent approach (generalized) semi-infinite programming relaxation and extension of MIP is studied for the identification of the unknown parameters which appear in the gene-environment (or target-environment) network models in the case of uncertainty is included (see [29, 30, 31, 32, 40, 42, 43, 45, 46] and the references inside).

Definition 2.3.5 *[84] Mixed-integer nonlinear programs (MINLP) are models of the general form*

$$z = \min \quad f(\boldsymbol{x}), \tag{2.19a}$$

$$s.t. \quad \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}, \tag{2.19b}$$

$$\boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \tag{2.19c}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is an objective function, and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^m$ is a constraint system.

In this definition, the functions f and $\mathbf{g}$ are assumed to be continuous and that $X := \{\mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$ is a compact set with $p \in \mathbb{N}_0$, $n \in \mathbb{N}$. This implies that f attains its minimum for some $\mathbf{x} \in X$ and therefore (2.19) is a well-defined problem. The question is how to actually solve a problem of the form (2.19) numerically [84].

The function f can be assumed as linear, without loss of generality. If not, we can introduce a new variable y and add the constraint $f(\mathbf{x}) \leq y$ to the constraint system (2.19b). Together with the new objective function min y we then obtain a problem that is equivalent to (2.19), but with a linear objective function. If $\mathbf{g}$ is differentiable and $p = 0$, then (2.19) is a pure nonlinear optimization problem, and techniques from constrained nonlinear optimization can be applied. If $\mathbf{g}$ fulfills further regularity assumptions, the Karush-Kuhn-Tucker (KKT) conditions provide necessary conditions for a solution to be (local) optimal; see [114]. These techniques originate from numerical analysis and yield only stationary points or local optima, if no further convexity assumption is made. Moreover, in the case of $p > 0$, they are not able to handle integrality restrictions on the variables. However, in case of a convex optimization problem, that is, if X is a convex set and f is a convex function, they are able to find a global optimum (see [84] and references therein).

For a general MINLP with a non-convex set X there are several methods described in the literature in order to relax (2.19) to a convex and continuous problem, such that a proven global optimum can be achieved at least for the relaxed problem [115, 116, 117]. For a solution of MINLP by relaxation and with branch-and-bound process, we refer to [84] and references therein.

CHAPTER 3

DYNAMICAL MODELS OF GENE-ENVIRONMENT NETWORKS

Dynamical systems play an important role in mathematical modeling. A huge number of phenomena from various fields of science and engineering were successfully described by using the powerful methods and techniques from this field.

In this chapter, the general concept of dynamical systems will be introduced briefly, then through the thesis we will concentrate on the dynamical modeling approach, based on systems of ordinary differential equations, to represent the dynamics of gene-environment regulatory systems. Differential equations are frequently used for modeling formalisms in mathematical biology. Ordinary differential equations offer a deterministic time and state continuous description of a given system. The evolution of the state $\mathbf{x} \in \mathbb{R}^n$ in time $t \in T$ is defined by the following function:

$$\boldsymbol{\varphi} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n, \quad \mathbf{x}(t) = \boldsymbol{\varphi}(t, \mathbf{x}(t_0)), \tag{3.1}$$

which is assumed to be the solution of an initial value problem given below

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad \mathbf{x}(t_0) = \mathbf{x_0}, \tag{3.2}$$

where $\mathbf{x_0} \in \mathbb{R}^n$ is a given state at a time t_0, $\mathbf{f} \in C^1(\mathbb{R}^n \to \mathbb{R}^n)$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ denotes the state whose components corresponds to states x_i of each component of the system. In gene regulatory networks, x_i $(i = 1, 2, \ldots, n)$ correspond to concentrations of network components [1]. In autonomous dynamic models, the states $\mathbf{x}$ depend on time, $\mathbf{x} = \mathbf{x}(t)$, and the set T describes a set of different time points. In time-discrete models, $T = \{t_0, t_1, \ldots, t_N\}$ contains a set of discrete time points, in time-continuous models, T is usually chosen to be the set of real numbers, $T = \mathbb{R}$.

The temporal behavior of a state $\mathbf{x}(t)$ of a system which is considered to model a regulatory network is given as a function $\boldsymbol{\varphi}(t, \mathbf{x}(0))$ of the initial state $\mathbf{x}(0)$ and the time t. Moreover, we assume that $\mathbf{x}(t)$ satisfies the following initial value problem

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad \mathbf{x}(t) \in D, \quad \mathbf{x}(0) = \mathbf{x_0}, \tag{3.3}$$

with an open set $D \subseteq \mathbb{R}^n$ and a function $\mathbf{f} \in C^1(D \to \mathbb{R}^n)$ [1]. The following theorem states the existence and uniqueness of the solution of a given initial-value problem.

Theorem 3.0.6 *[118] Let D be an open subset of $\mathbb{R}^n$ containing $\mathbf{x_0}$ and assume that $\boldsymbol{f} \in C^1(D)$. Then there exists an $a > 0$ such that the initial value problem $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$ with initial value $\mathbf{x}(0) = \mathbf{x_0}$ has a unique solution $\mathbf{x}(t)$ on the time interval $[-a, a]$.*

Definition 3.0.7 *[118] A dynamical system on D is a C^1-map*

$$\boldsymbol{\varphi} : \mathbb{R} \times D \to D, \tag{3.4}$$

where D is an open subset of $\mathbb{R}^n$, and if $\boldsymbol{\varphi}_t(\mathbf{x}) := \boldsymbol{\varphi}(t, \mathbf{x}(t))$ then

1. $\boldsymbol{\varphi}_0(\mathbf{x}) = \mathbf{x}$, *for all* $\mathbf{x} \in D$, *and*
2. $(\boldsymbol{\varphi}_t \circ \boldsymbol{\varphi}_s)(\mathbf{x}) = \boldsymbol{\varphi}_{t+s}(\mathbf{x})$ *for all* $s, t \in \mathbb{R}$ *and* $\mathbf{x} \in D$.

The relation between a dynamical system and an initial value problem can be stated in such a way that [1]:

When $\boldsymbol{\varphi}(t, \mathbf{x}(t))$ is a dynamical system defined on $D \subseteq \mathbb{R}^n$, then

$$\mathbf{f}(\mathbf{x}) = \frac{d}{dt}\boldsymbol{\varphi}(t, \mathbf{x})|_{t=0} \tag{3.5}$$

defines a C^1-vector field on D, and $\boldsymbol{\varphi}(t, \mathbf{x_0})$ solves the initial value problem in (3.3) for each $\mathbf{x_0} \in D$. $\boldsymbol{\varphi}(t, \mathbf{x_0})$ can also be considered as the motion of the set D through the state space, so it can be called as *flow* of the differential equation [118].

Dynamical systems are divided to two major classes as linear and nonlinear depending on the form of the right hand side function in (3.3) with respect to $\mathbf{x}$. The results for the stability of fixed points of linear systems can be transferred to analyze the stability of fixed points of nonlinear systems [118]. In the following we present the definitions of an equilibrium point and its stability which is important for analyzing the long-term dynamic behavior of a dynamical system.

Definition 3.0.8 *[118] A point* $\mathbf{x}^* \in \mathbb{R}^n$ *is called an equilibrium point of a system* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ *if* $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$*. An equilibrium point* $\mathbf{x}^*$ *is called a hyperbolic equilibrium point if none of the eigenvalues of the Jacobian matrix* $\mathbf{J_f}(\mathbf{x}^*)$ *has zero real part. The linear system* $\dot{\mathbf{x}} = \mathbf{J_f}(\mathbf{x}^*)\mathbf{x}$ *is called the linearization of (3.3) at* $\mathbf{x}^*$*.*

In the case that $\mathbf{x}^*$ is an equilibrium point of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ and $\boldsymbol{\varphi}_t$ is the flow of the system, then $\boldsymbol{\varphi}_t(\mathbf{x}^*) = \mathbf{x}^* \ \forall t \in \mathbb{R}$, and $\mathbf{x}^*$ is called a *fixed point* or a *steady state*.

Definition 3.0.9 *[118] Let* $\boldsymbol{\varphi}_t(\mathbf{x})$ *represent the flow of the differential equation* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \ \forall t \in \mathbb{R}$*. An equilibrium point* $\mathbf{x}^*$ *of this system is stable if* $\forall \epsilon > 0 \ \exists \ \delta > 0$ *such that* $\forall \mathbf{x} \in N_\delta(\mathbf{x}^*)$ *and all* $t \geq 0$*,* $\boldsymbol{\varphi}_t(\mathbf{x}) \in N_\epsilon(\mathbf{x}^*)$ *holds.*

The equilibrium point $\mathbf{x}^*$ *is unstable if it is not stable, and asymptotically stable if* $\exists \ \delta > 0$ *such that* $\lim_{t\to\infty} \boldsymbol{\varphi}_t(\mathbf{x}) = \mathbf{x}^* \ \forall \mathbf{x} \in N_\delta(\mathbf{x}^*)$*.*

3.1 Time-Continuous Model Class of Gene-Environment Networks

Differential equations are commonly used for modeling formalisms in mathematical biology. The main reasons for that are: modeling of regulatory interactions by differential equations can provide a more accurate understanding and explanation of the physical systems, and there exist many well developed approaches like dynamical systems theory for analyzing these models and capturing their dynamical behavior. Also, considering that the biological systems developed in continuous time, it is preferred to use systems of differential equations which can allow to do instantaneous changes on their right-hand sides [119]. In general, a differential relation among n variables of gene networks is represented by the following equation

$$\frac{dx_i}{dt} = f_i(\mathbf{x}) \quad (i = 1, 2, \ldots, n), \tag{3.6}$$

where each function $f_i : \mathbb{R}^n \to \mathbb{R}$ is nonlinear and the vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ represents the positive concentrations of proteins, mRNAs, or small components.

Firstly proposed dynamical model for gene regulation is given by Chen et al. [27] with a system of linear differential equations having a constant coefficient matrix, then, in [120, 121, 122], a discretized linear model is used to infer the structure of regulatory networks. Although linear models may not be enough to represent the whole dynamics of the system, they are

still frequently used to infer regulations between cell components (we refer the following references [122, 123, 124, 125, 126, 127]). The general aim of mentioned works and the other related ones was the inference of the structure of the interaction graph rather than capturing the dynamic behavior. In [128], it was argued that regulation functions can usually be well approximated by their linearization. Also in [129], linear models were used together with a discussion of its advantages and disadvantages. In the problem of parameter estimation from experimental data, linear models provide several advantages [128]. These models can be solved analytically and the minimization of the sum of squared errors between measurements and model predictions is an optimization problem having a quadratic objective function. On the other hand, linear models cannot represent the whole of dynamic behaviors observed in regulatory networks like periodic behavior, multi stationarity and hysteresis (see [1] for examples and further details). In [1], it is observed that linear models have only one single isolated equilibrium point which can be stable or unstable. If the equilibrium point is stable, then it is globally stable and convergence to that point is reached from all initial conditions. Moreover, a linear system is unstable meaning that it does not omit unbounded solutions. More detailed information about qualitative behavior of linear models, analysis of limit sets and stability of equilibrium points of nonlinear models can be found in [1].

In the literature, the firstly introduced time-continuous models to represent the dynamical behavior of gene-environment networks were given by the following systems of ordinary differential equations (ODEs) of the time-autonomous form

$$\dot{\mathbb{E}} = \mathbb{F}(\mathbb{E}), \tag{3.7}$$

where $\mathbb{E} = (\mathbb{E}_1, \mathbb{E}_2, \ldots, \mathbb{E}_d)^T$ is the d-vector of positive concentration levels of proteins (or mRNAs, or small components) and of certain levels of the environmental factors. Here, $\mathbb{E} = \mathbb{E}(t)$ where the time t is in the interval I where $I = (a, b) \subseteq \mathbb{R}$. The first n entries of the vector $\mathbb{E}$ refer to the genes, whereas the remaining part, with $d - n$ components, refers to the environmental factors. $\dot{\mathbb{E}}\,(= \frac{d\mathbb{E}}{dt})$ represents a continuous change in the gene-expression data, and $\mathbb{F}_i : \mathbb{R}^d \to \mathbb{R}$ are nonlinear coordinate functions of $\mathbb{F}$,i.e., $\mathbb{F} = (\mathbb{F}_1(\mathbb{E}), \mathbb{F}_2(\mathbb{E}), \ldots, \mathbb{F}_d(\mathbb{E}))$ [10, 27, 35, 36, 55, 58, 59]. The estimation of parameters associated and contained in the definition of $\mathbb{F}$ is studied by considering the experimental data vectors $\bar{\mathbb{E}}$ of these levels which are obtained from microarray experiments and from environmental measurements at the sample times. The vectors $\bar{\mathbb{E}}$ are just approximate data to the actual states $\mathbb{E}$ at the sample times of the experiments, so they may contain some errors, noise and uncertainties coming from the

measurements [29, 30, 43, 44, 45, 46]. Here, $\mathbb{E}(t_0) = \mathbb{E}_0$ denotes the initial values, where $\mathbb{E}_0 = \bar{\mathbb{E}}_0$. Moreover, $E_i(t)$ stands for the gene-expression level (concentration rate) of the i^{th}-gene at time t, and $E_i(t)$ denotes anyone of the first n coordinates in the d-vector $\mathbb{E}$ of genetic and environmental states. We write $G := \{1, 2, \ldots, n\}$ for the set of genes [84].

There is a collection of improved types of Eq. (3.7) representing the dynamical system on the gene-expressions and having the following forms given in [10, 11, 25, 27, 28, 35, 36, 37, 41, 42, 59, 130]:

(i) Chen et al. [27] proposed the first time-continuous model consisting of system of first order ODEs

$$\dot{\mathbf{E}} = \mathbf{M}\mathbf{E}, \tag{3.8}$$

to model time-series gene expression patterns, where $\mathbf{M}$ is an $(n \times n)$-constant matrix as a transition matrix representing regulatory interactions for both genes and proteins, and $\mathbf{E}$ is the $(n \times 1)$-vector representing the expression level of individual genes. Later on, Hoon et al. [58, 59] studied another continuous model similar to this model, but they consider only mRNA concentrations and AIC was used to identify the places and number of nonzero parameters in the coefficient matrix $\mathbf{M}$. Sakamoto and Iba [35] suggest a more flexible model as

$$\dot{E}_j = F_j(E_1, E_2, \ldots, E_n)^T \quad (j = 1, 2, \ldots, n), \tag{3.9}$$

where F_j are functions of $\mathbf{E} = (E_1, E_2, \ldots, E_n)^T$ determined by genetic programming and least-squares methods.

(ii) The above models were refined and new ideas are introduced by [10, 36, 37, 41, 131] for the improvement. Gebert et al. [10] use a constant gene interaction matrix in the model $\dot{\mathbf{E}} = \mathbf{M}\mathbf{E}$ and use discrete least-squares approximation [132] for the estimation of parameters appearing in the regulatory relations. In [36], authors suggested the following nonlinear or, say, quasi-linear model

$$\dot{\mathbf{E}} = \mathbf{M}(\mathbf{E})\mathbf{E}. \tag{3.10}$$

with its deterministic matrix multiplicative form. This multiplicative form becomes an iterative multiplicative one in the corresponding time-discrete dynamics which is presented in the next section.

In this formulation, the interaction matrix $\mathbf{M}$ depends on the current metabolic state $\mathbf{E}$. In this way, nonlinear interactions between network variables were taken into account but the solution space is restricted by imposing bounds on the number of regulating factors for each gene. The reason behind was to identify a unique regulatory network.

This dynamical system refers to the n genes and their interaction alone so that the matrix $\mathbf{M}$ is an $(n \times n)$-matrix with entries as functions of polynomials, exponential, trigonometric, splines or wavelets containing some parameters to be optimized. In the matrix $\mathbf{M}$, each row and column corresponds to one gene in the genetic network. The value of each entry of the matrix refers to the interaction between genes. If these entries have zero value, this means that there is no interaction between the genes. The smaller the absolute value of the matrix entries, the less is the influence or interaction between genes [10].

(iii) In [25], an extended version of the model given by Eq. (3.10) is derived to emphasize the nonlinear interactions with the environment. The model is represented as

$$\dot{\mathbf{E}} = \mathbf{F}(\mathbf{E}), \tag{3.11}$$

where $\mathbf{F}(\mathbf{E})$, given as $\mathbf{F} = (F_1, F_2, \ldots, F_n)^T$, consists of a sum of quadratic functions (also covers constant and linear case)

$$F_j(\mathbf{E}) = f_{j,1}(E_1) + f_{j,2}(E_2) + \ldots + f_{j,n}(E_n) \quad (j = 1, 2, \ldots, n). \tag{3.12}$$

Yılmaz [25] also added affine linear shifts terms to this model and extend it.

(iv) To keep the recursive iteration idea, that is presented in [37], by these shifts, Eq. (3.10) is reconstructed below from the following system that includes an affine addition [11, 130]:

$$\dot{\mathbf{E}} = \mathbf{M}(\mathbf{E})\mathbf{E} + \mathbf{C}(\mathbf{E}). \tag{3.13}$$

Here, $\mathbf{C}(\mathbf{E})$ is an additional column vector representing environmental perturbations or contributions and provides a more accurate data fitting (cf. [25, 41] for the case of a constant $\mathbf{C}$). The shift term $\mathbf{C}(\mathbf{E})$ does not need to reveal $\mathbf{E}$ as a factor while $\mathbf{M}(\mathbf{E})\mathbf{E}$ reveals $\mathbf{E}$ as a factor, but it can be, e.g., exponential, trigonometric, also splines [43, 29]. In the extended model [11, 25, 39, 41, 42, 130] represented by Eq. (3.13), the dimension of the vector $\mathbf{E}$ is increased to $n + m$ by considering the m-vector $\check{\mathbf{E}}(t) =$

$(\check{E}_1(t), \check{E}_2(t), \ldots, \check{E}_m(t))^T$, which represents m environmental factors affecting the gene-expression levels and their variation. To represent the weights of the effect of the j^{th}-environmental factor $\check{E}_j$ on the gene-expression data E_i, the $(n\times m)$-weight matrix $\check{\mathbf{M}}(\mathbf{E})$ is introduced so that the vector $\mathbf{C}(\mathbf{E})$ can be written as $\mathbf{C}(\mathbf{E}) = \check{\mathbf{M}}(\mathbf{E})\check{\mathbf{E}}$, where

$$\check{\mathbf{M}}(\mathbf{E}) = \begin{pmatrix} c_{11}(\mathbf{E}) & \cdots & c_{1m}(\mathbf{E}) \\ \vdots & \ddots & \vdots \\ c_{n1}(\mathbf{E}) & \cdots & c_{nm}(\mathbf{E}) \end{pmatrix} \tag{3.14}$$

is called as the gene-environment matrix. Its entries c_{ij} are the weights. Therefore, the gene-environment network described by the dynamic equation in (3.13) becomes

$$\dot{\mathbf{E}} \;=\; \mathbf{M}(\mathbf{E})\mathbf{E} + \check{\mathbf{M}}(\mathbf{E})\check{\mathbf{E}}. \tag{3.15}$$

Finally, the extended initial value problem can be written in a multiplicative form as follows:

$$\dot{\mathbb{E}} = \mathbb{M}(\mathbb{E})\mathbb{E}, \qquad \mathbb{E}_0 = \mathbb{E}(t_0) = \begin{bmatrix} \mathbf{E}_0 \\ \check{\mathbf{E}}_0 \end{bmatrix}, \tag{3.16}$$

where

$$\mathbb{E} := \begin{bmatrix} \mathbf{E} \\ \check{\mathbf{E}} \end{bmatrix}, \;\; \mathbb{M}(\mathbb{E}) := \begin{pmatrix} \mathbf{M}(\mathbf{E}) & \check{\mathbf{M}}(\mathbf{E}) \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{3.17}$$

are an $(n + m)$-vector and $(n + m) \times (n + m)$-matrix, respectively. The other versions of the extended gene-environment network in Eq. (3.13) are studied in [29, 42, 43].

Another version of the extended gene-environment network in Eq. (3.13) is studied in [29, 38, 42, 43] by splitting the shift $\mathbf{C}(\mathbf{E})$ as $\mathbf{C}(\mathbf{E}) = \mathbf{W}(\mathbf{E})\check{\mathbf{E}} + \mathbf{V}(\mathbf{E})$ and increasing the dimension to $m + 2n$. Here, $\mathbf{V}(\mathbf{E})$ stands for all the cumulative effects of the environmental items and $\check{\mathbf{E}}$ is a specific m-vector, representing the levels of the m-environmental factors.

As a system of ordinary differential equations, Eq. (3.8) - Eq. (3.13), hence, Eq. (3.16) are all autonomous, which means that the right-hand side depends on the state $\mathbf{E}$ only, but not on time t. This implies that trajectories do not cross themselves (see [36] for details).

The models given by the continuous dynamical equations in **(i)**, **(ii)**, **(iii)** and **(iv)** can be written in general as ([29, 42, 43, 46])

$$\dot{\mathbb{E}} = \mathbb{M}(\mathbb{E})\mathbb{E}, \tag{3.18}$$

with the initial value $\mathbb{E}_0 = \mathbb{E}(t_0)$, where $\mathbb{E}$ and $\mathbb{E}_0$ are $(d \times 1)$-vectors. The $(d \times d)$-matrix $\mathbb{M}(\mathbb{E})$ has entries which contain parameters to be estimated, see [50, 76]. The entries of $\mathbb{M}(\mathbb{E})$, which can be polynomial, trigonometric, exponential, logarithmic, hyperbolic, spline, etc., represent the growth, cyclicity or other kinds of changes in the genetic or environmental concentration rates that is supposed by any kind of a priori information, observation or assumption [36]. The form of system in (3.18) allows a time-discretization such that the dynamics is given by a step-wise matrix multiplication. This recursive property is an important advantage of the this form in terms of algorithmic stability analysis (see [11, 130] and cited references).

3.2 Corresponding Time-Discrete Models

The discretization process can be defined as transferring continuous models and equations to discrete counterparts. A numerical solution is generated by simulating the behavior of a system that is expressed by ordinary differential equations. It is initiated at the starting time point t_0 with a given initial value $\mathbf{E}_0$ and approximates the solution at discrete time points during the given time interval. The approximate value of the solution at a state is generated from the value at the previous states iteratively. Here, the main concerns are the stability and precision of the simulated approximate results [11, 133]. By using discretization methods, we can discretize a continuous process and so that we can obtain the approximate solutions $E_0, E_1, \ldots, E_{N-1}$ at discrete time points $t_0, t_1, \ldots, t_{N-1}$ and then compare them with the given experimental values $\bar{E}_0, \bar{E}_1, \ldots, \bar{E}_{N-1}$. The comparison of Euler's and Runge-Kutta methods, as discretization methods (schemes), is studied in [133] for a concrete example of system of differential equations and stated that Euler's scheme produce unstable results compared with the exact solution. It is slow and inaccurate. Euler's method is the simplest discretization scheme and based on 1^{st}-order Taylor series expansion, so it is an approximation by a straight line [11, 133].

When numerical methods are used for the approximate solutions of ordinary differential equations, rounding errors and truncation errors can be faced with because of finite precision of floating-point arithmetic and number of iterations. Euler's method approximate the derivative, appearing in ordinary differential equations, at the beginning of each subinterval $[t_k, t_{k+1}]$ of the discretized time domain while Runge-Kutta methods use midpoints [11]. Runge-Kutta methods provide more advantage in stability, accuracy and programmability.

Moreover, Runge-Kutta methods are more sensitive to infinitesimal numerical changes [141]. These methods has geometric meaning as the "average of slopes" and they have both explicit and implicit versions. Explicit Runge-Kutta methods can be derived by using Taylor series expansion of order p to produce a method of order p (we refer to [141] and cited references for more details). Euler method can also be considered as 1^{st}-order explicit Runge-Kutta method.

The general formulation of the *explicit Runge-Kutta methods with ν-stages (ν-slopes)* applied to the system of ordinary differential equations of the form $\dot{x}(t) = f(x(t))$ in (3.6), is given as follows:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \sum_{i=1}^{\nu} \omega_i K_i, \quad \text{where} \\ K_i &= hf(t_k + c_i h, x^{(k)} + \sum_{j=1}^{i-1} a_{ij} K_j), \end{aligned}$$

with arbitrary coefficients c_i, a_{ij} and ω_i ($i = 1, 2, \ldots, \nu$; $j = 1, 2, \ldots, i-1$) and $c_1 = 0$.

In [11, 25, 29, 41, 42, 43, 46, 130], Euler's method and 2^{nd}-order Heun's method (as a 2^{nd}-order Runge-Kutta method or 2-stage Runge-Kutta method) are applied for the time-discrete dynamics of gene-environment (target-environment) regulatory systems listed in Section 3.1 together with the derived matrix algebra correspondingly. For the most general form of gene-environment (target-environment) network model expressed in Eq. (3.18), Euler's method is applied to discretize the time-continuous process as follows (see [11, 25, 38, 41, 130] and references therein):

$$\dot{\mathbb{E}}^{(k)} := \frac{\mathbb{E}^{(k+1)} - \mathbb{E}^{(k)}}{h_k} = \mathbb{M}(\mathbb{E}^{(k)})\mathbb{E}^{(k)}, \tag{3.19}$$

$$\Leftrightarrow \mathbb{E}^{(k+1)} = (\mathbb{I} + h_k \mathbb{M}(\mathbb{E}^{(k)}))\mathbb{E}^{(k)}, \tag{3.20}$$

where $\dot{\mathbb{E}}(t_k) \approx \dot{\mathbb{E}}^{(k)}$, for all $k \in \mathbb{N}_0$, $h_k = t_{k+1} - t_k$ is the step-size and $t_k < t_{k+1}$.

Hence, corresponding time-discrete dynamics is obtained by the following equation

$$\mathbb{E}^{(k+1)} = \mathbb{M}^{(k)}\mathbb{E}^{(k)} \quad (k \in \mathbb{N}_0). \tag{3.21}$$

The next state can be iteratively generated from the previous states approximately as follows

$$\widehat{\mathbb{E}}^{(k)} (:= \mathbb{E}^{(k)}) = \mathbb{M}^{(k-1)}(\mathbb{M}^{(k-2)} \cdots (\mathbb{M}^{(1)}(\mathbb{M}^{(0)}\bar{\mathbb{E}}^{(0)}))) \quad (k \in \mathbb{N}_0), \tag{3.22}$$

for a given initial value $\bar{\mathbb{E}}^{(0)} (= \mathbb{E}^{(0)})$, where $\bar{\mathbb{E}}^{(0)}, \bar{\mathbb{E}}^{(1)}, \ldots, \bar{\mathbb{E}}^{(N-1)}$ denotes the provided experimental values and $\widehat{\mathbb{E}}^{(0)}, \widehat{\mathbb{E}}^{(1)}, \ldots, \widehat{\mathbb{E}}^{(N-1)}$ refers to the approximated (estimated) values. This multiplicative form provide an important advantage both computationally and analytically.

For the derivation of iterative formula of 2^{nd}-order Heun's method for the dynamics of gene-environment networks and corresponding matrix algebra of both methods, we refer to [11, 25, 29, 38, 41, 42, 43, 46, 130, 134]. Higher order Runge-Kutta methods, namely 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method, are newly derived and studied for the dynamics of gene-environment networks in the following chapter together with their matrix algebra.

3.2.1 Stability Analysis

The mathematical meaning of stability of dynamical systems is expressed by stationary (equilibrium) points. In analysis, it means that being in some sufficiently small neighbourhood of the equilibrium, the system never escape. In studied dynamic model class for gene-environment networks and its discretized versions, the step-wise multiplicative form of the time-discrete system becomes evaluated and interpreted as the boundedness of the dynamics [37, 135].

Genomic stability in molecular biology refers to the ability of an organism to repair physical and chemical damages and changes of the genome. In the model class presented in Section 3.1, gene-expressions are considered to be stable if there is not any important and sudden changes along the time of transition process from one stable metabolic state to another [37].

The fundamental definitions and theorems stated in the following are related with the stability of our considered time-continuous dynamic model class and its time-discretized version.

Definition 3.2.1 *[119] A point $\mathbb{E}^* \in \mathbb{R}^d$ is called an equilibrium point of system*

$$\dot{\mathbb{E}} = f(t, \mathbb{E}), \tag{3.23}$$

where $(t, \mathbf{E}) \in \mathbb{R} \times \mathbb{R}^d$ if $f(t, \mathbf{E}^*) = 0 \; \forall t \in \mathbb{R}$. An equilibrium $\mathbb{E}^*$ of the system (3.23) is called stable (in Lyapunov sense) if for every $\varepsilon > 0$ there exists a $\delta = \delta(\varepsilon) > 0$ such that at time $t = t_0$ it satisfies

$$\left\| \mathbb{E}(t_0) - \mathbb{E}^* \right\| < \delta,$$

and for all $t > t_0$ it holds

$$\left\| \mathbb{E}(t) - \mathbb{E}^* \right\| < \varepsilon.$$

An algorithmic method in order to study stability of time-discrete versions of the models is firstly given by [136]. Together with the relation between the stability of time-discrete and time-continuous systems. For some time-discretization methods, this approach is studied in [11, 130, 134, 136]. The following theorems states the strong relation between the stability of time continuous model $\dot{\mathbb{E}} = \mathbb{M}(\mathbb{E})\mathbb{E}$ and the stability of time-discrete system $\mathbb{E}^{(k+1)} = \mathbb{M}^{(k)}\mathbb{E}^{(k)}$, obtained by Euler and Runge-Kutta discretizations.

Theorem 3.2.2 *[136] Let the map $\mathbb{E} \mapsto \mathbb{M}(\mathbb{E})$ be Lipschitzian. If the Eulerian time-discrete system $\mathbb{E}^{(k+1)} = \mathbb{M}^{(k)}\mathbb{E}^{(k)}$ $(k \in \mathbb{N}_0)$, $\mathbb{E}^{(0)} \in \mathbb{R}^d$, with some appropriate $h_{max} > 0$ being given, is stable for any values $h_k \in [0, h_{max}]$, then the time-continuous system $\dot{\mathbb{E}} = \mathbb{M}(\mathbb{E})\mathbb{E}$ is also stable.*

For the proof we refer to [136].

Theorem 3.2.3 *[42] Let the map $\mathbb{E} \mapsto \mathbb{M}(\mathbb{E})$ be Lipschitzian. If the Runge-Kutta time-discrete system $\mathbb{E}^{(k+1)} = \mathbb{M}^{(k)}\mathbb{E}^{(k)}$ $(k \in \mathbb{N}_0)$, $\mathbb{E}^{(0)} \in \mathbb{R}^d$, with some appropriate $h_{max} > 0$ being given, is stable for any values $h_k \in [0, h_{max}]$, then the time-continuous system $\dot{\mathbb{E}} = \mathbb{M}(\mathbb{E})\mathbb{E}$ is also stable.*

For the proof we refer to [42].

3.3 Identification of Parameters

Modeling and prediction of gene-expression patterns are characterized by two different kinds of variables: the gene-expression levels (concentration rates), and the variables ("parameters") that represent the dynamics of the gene-expression levels (changes in the concentration rates). Environmental effects can also be included likewisely. In this "duality", one class of variables is considered as parameters perturbed and the remaining class of variables observed how to reply to those perturbations [37, 42, 45, 46, 130]. In this way, the whole learning problem is expressed in bilevel problems [137] of decision and optimization. In order to have a deep understanding about state and variation of genetic (and environmental) patterns we use *matrices* which play a dual part. These matrices are called network matrices which are important for testing the *goodness of data fitting* and *prediction* and they are identified by least-squares

(or maximum likelihood) estimation. They represent the dynamics of the network (system), with behaviors of expansion, rest, contraction or cyclicity where all of these behaviors can be comprised by *stability* vs. *instability*, respectively, and they mean time step-wise or time-continuous. The discrete sequences (orbits) obtained by step-wise matrix multiplication have been studied by [1, 11, 36, 37, 39, 42, 46, 55, 130, 138] algebraically by the algorithm of Brayton and Tong [136]. This algorithm constructs and analyzes a sequence of compact neighbourhoods of the zero point in the state space of gene-expression (and environmental) levels, and they are chosen as bounded polyhedra; i.e., they can be encoded and dynamically defined by the finite sets of their vertices, where the construction principle consists of finite numbers of matrix multiplications in each iterations. We refer to [37, 42, 46, 130] and references therein for further details.

Let us consider the general form of the gene-environment network given in Section 3.1 that is represented by the dynamical equation $\dot{\mathbb{E}} = \mathbb{M}(\mathbb{E})\mathbb{E}$, where the entries of matrix $\mathbb{M}(\mathbb{E})$ contain parameters to be estimated [50, 76]. The entries of $\mathbb{M}(\mathbb{E})$, which can be polynomial, trigonometric, exponential, but otherwise logarithmic, hyperbolic, spline, etc., represent the growth, cyclicity or other kinds of changes in the genetic or environmental concentration rates that we suppose by any kind of a priori information, observation or assumption [10, 36].

Two different levels of the problem concerning the parametrized entries of the matrices can just be distinguished: optimization and stability analysis, both of them constituting bilevel problems (see [10, 29, 36, 37, 42, 43, 46, 138] and references therein).

The first step is the *optimization problem* of approximation with respect to squared errors (say, *discrete least-squares approximation*):

$$\min_{\mathbf{y}} \sum_{\kappa=0}^{N-1} \left\| \mathbb{M}_y(\bar{\mathbb{E}}^{(\kappa)})\bar{\mathbb{E}}^{(\kappa)} - \dot{\bar{\mathbb{E}}}^{(\kappa)} \right\|_2^2. \tag{3.24}$$

where $\mathbf{y}$ is the vector of a subset of all the parameters, N is the number of measurements and the $\dot{\bar{\mathbb{E}}}^{(\kappa)}$ are the difference quotients based on the κ^{th}experimental data $\bar{\mathbb{E}}^{(\kappa)}$ with interval lengths $\bar{h}_\kappa := \bar{t}_{\kappa+1} - \bar{t}_\kappa$ between neighbouring samplings. Note that, forward and central difference approximations are the common choices for approximating $\dot{\bar{\mathbb{E}}}^{(\kappa)}$ as it is described below:

$$\dot{\bar{\mathbb{E}}}^{(\kappa)} := \begin{cases} (\bar{\mathbb{E}}^{(\kappa+1)} - \bar{\mathbb{E}}^{(\kappa)})/\bar{h}_\kappa, & \text{if} \quad \kappa \in \{0, 1, \ldots, N-1\}, \\ (\bar{\mathbb{E}}^{(N)} - \bar{\mathbb{E}}^{(N-1)})/\bar{h}_\kappa, & \text{if} \quad \kappa = N. \end{cases} \tag{3.25}$$

For the equidistant step-size, $\bar{h}_\kappa \equiv c$ ($c > 0$), $\dot{\bar{\mathbb{E}}}^{(\kappa)} := (\bar{\mathbb{E}}^{(\kappa+1)} - \bar{\mathbb{E}}^{(\kappa-1)})/2c$ is a common choice [28, 29, 36, 46].

In problem (3.24), the Euclidean norm is referred $\|\cdot\|_2$ but the Chebychev norm $\|\cdot\|_\infty$ can also be used in the case where uncertainty is included. The least-squares methods of linear and nonlinear regression are used to estimate the vector $\mathbf{y}$ of a first part of the parameters to fit the set of given experimental data and to characterize the statistical properties of estimates.

The second step is the *stability of the dynamics* investigated with respect to the remaining parameters, as mentioned in the beginning of this section. For this a combinatorial algorithm based on polyhedra sequences is employed to detect the regions of stability and instability [28, 36, 37, 46].

Since real-world gene-environment networks are very large, for practical reasons one has to simplify them by diminishing the number of arcs [42, 46]. Let us present the basic idea of such a *rarefication*:

Generally used microarray gene-expression data is very noisy. This can lead the network to have many artificial and low weighted connections. Since there are usually a few genes in the network which are involved strongly in regulations, then those genes have high outdegree values. In order to limit these values, a bound is introduced for every gene in the network [28] according to which vertices are important and how that expresses itself in the bound. The solutions which does not satisfy the bound are rejected.

Also, in provided data sets, the number of genes are bigger than the time points which results the system to be under determined and so having many optimal solutions. Therefore, a restriction should be brought into the solution space. For this purpose, some ideas are proposed [28, 36, 37] like the off-diagonal entries of network matrix $\mathbf{M}$ to be nonnegative. The underlying reason is that the degradation of gene products is supposed to be proportional to the concentration of the gene product itself and does not depend on any other variables [28].

Beyond the biological meaning, the choice of constraints and their bounds are related with the *decision making*, multi-criteria optimization, *rarefication* and *regularization*. The bounds to be added to the problem in (3.24) will form it to the mixed integer problem given in Subsection 4.3.1. These bounds are imposed to regularize and rarefy the gene-network in a way of selecting the most important and meaningful elements in the network. One aspect of doing

this is to put bounds on the objective function of the considered optimization problem and another aspect is to put bounds on the constraints of the problem. The selection the bound is about decision making and related with how we want to rarefy the given network (system). It is also a matter of parametric variation. Each value of a bound should provide a solution on an *efficiency frontier* (in the case of parametric variation of one bound) or an efficiency surface (in the case of parametric variation of several bounds) (from multi-objective optimization). Then, we can select the right or best element of the efficiency frontier, as the solution, according to statistical comparison and performance criteria [139, 140]. That performance and comparison criteria from statistics, but also numerical mathematics, can be employed to select an "optimal" solution from the efficiency barrier (efficiency frontier or efficiency surface) (see [79, 82, 139, 140] and references therein).

Complex regulatory systems generally contain a large number of interconnected components and the target-environment and gene-environment regulatory network is highly structured with multiple interactions among many different clusters. It may be necessary to reduce the number of branches of the these regulatory networks for computational purposes. For this case, bounds on the indegrees of the nodes (clusters) can reduce the complexity of the model. Binary constraints can be used to decide whether or not there is a connection between pairs of clusters [44]. Imposing these additional constraints to the objective function of the regression problem initiated above in Eq. (3.24), we obtain a *mixed integer optimization problem* which corresponds to our network rarefication that will be formulated in details in Chapter 4, Subsection 4.3.1.

However, binary constraints are very strict and they can even destroy the connectivity of the regulatory network in some cases. In order to not to be faced with these difficulties, the binary constraints can be replaced by more flexible continuous constraints leading to a further relaxation in terms of continuous optimization (see [29, 32, 30, 31, 40, 42, 43, 45, 46] and related references therein).

CHAPTER 4

NEW DISCRETIZATION SCHEMES FOR GETTING THE TIME DISCRETE MODELS WITH APPLICATIONS

To approximate the time-continuous dynamical models of regulatory networks listed in Chapter 3, Section 3.1, various discretization schemes can be used to obtain the numerical solution at a discrete set of points in time. As it is explained in Chapter 3, Section 3.2, it is important to choose the appropriate method to be applied for this purpose. Firstly, Euler's method was used in the time-discretization for the gene-expression patterns; it has been seen that Euler's method is slow and inaccurate (see [133] for further information). Then, Runge-Kutta methods were introduced in [134] and, specifically, the 2^{nd}-order Heun's method was studied in [11, 130] known as the simplest Runge-Kutta approach. In terms of rounding error and truncation error, the choice of the method in the numerical derivations plays an important role. Comparing with Euler's method, Runge-Kutta methods have advantages in truncation error, and in stability which is closer to the stability of the time-continuous model, and in implementation [141].

In this chapter of the thesis, we newly study the 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method, as *explicit higher order Runge-Kutta methods*, for the discretization of the time-continuous models to improve the rate of convergence and accuracy. These new results are recently published in the articles [45, 84, 85] of the author together with the application of 3^{rd}-order Heun's method on a set of artificial data. In addition to these published works, this thesis also includes the following original results:

The application of 4^{th}-order classical Runge-Kutta method is newly studied in this thesis within a comparison process of all these four numerical schemes tested on the same artificial data set. The performance of these methods are investigated with different step-sizes and also

their sensitivity with respect to a perturbation is tested. Moreover, as a second numerical example, a real-world application is examined for all these methods on a real data set. Analysis of considered real data and selection of constraints for the corresponding optimization problem is studied both biologically and mathematically in order to perform a parameter estimation. All obtained new results of these two illustrative examples are presented with the help of the figures. Moreover, detailed "discussions" on the numerical results are done after each performed analysis and also in Subsubsection 4.3.1.5 and Subsubsection 4.3.2.4.

4.1 Formulation of the Numerical Schemes

In the most general model of gene-environment network that is given by Eq. (3.18), we newly apply, by this thesis, the 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method respectively and newly formulate our time-discrete models and derive their matrix algebra correspondingly as presented in the following.

1. **Using 3rd-order Heun's method:**

$$\begin{aligned}
\mathbb{E}^{(k+1)} &= \mathbb{E}^{(k)} + \frac{h_k}{4}(k_1 + 3k_3), \\
k_1 &= \mathbb{M}(\mathbb{E}^{(k)})\mathbb{E}^{(k)}, \\
k_2 &= \mathbb{M}(\mathbb{E}^{(k)} + \frac{h_k}{3}k_1)(\mathbb{E}^{(k)} + \frac{h_k}{3}k_1), \\
k_3 &= \mathbb{M}(\mathbb{E}^{(k)} + \frac{2h_k}{3}k_2)(\mathbb{E}^{(k)} + \frac{2h_k}{3}k_2).
\end{aligned} \tag{4.1}$$

$$\begin{aligned}
\mathbb{E}^{(k+1)} &= \mathbb{E}^{(k)} + \frac{h_k}{4}\mathbb{M}(\mathbb{E}^{(k)})\mathbb{E}^{(k)} \\
&\quad + \mathbb{M}(\mathbb{E}^{(k)} + \frac{2h_k}{3}\mathbb{M}(\mathbb{T}^{(k)})\mathbb{T}^{(k)}) \\
&\quad \times \{\frac{3h_k}{4}\mathbb{E}^{(k)} + \frac{h_k^2}{2}\mathbb{M}(\mathbb{T}^{(k)})\mathbb{E}^{(k)} \\
&\quad + \frac{h_k^3}{6}\mathbb{M}(\mathbb{T}^{(k)})\mathbb{M}(\mathbb{E}^{(k)})\mathbb{E}^{(k)}\}.
\end{aligned} \tag{4.2}$$

Then, we get the time-discrete equation as

$$\mathbb{E}^{(k+1)} = \mathbb{M}^{(k)}\mathbb{E}^{(k)}, \tag{4.3}$$

where

$$\begin{aligned}\mathbb{M}^{(k)} &:= \mathbb{I} + \frac{h_k}{4}\mathbb{M}(\mathbb{E}^{(k)}) + \mathbb{M}(\mathbb{E}^{(k)} + \frac{2h_k}{3}\mathbb{M}(\mathbb{T}^{(k)})\mathbb{T}^{(k)}) \\ &\quad \times\{\frac{3h_k}{4}\mathbb{I} + \frac{h_k^2}{2}\mathbb{M}(\mathbb{T}^{(k)}) + \frac{h_k^3}{6}\mathbb{M}(\mathbb{T}^{(k)})\mathbb{M}(\mathbb{E}^{(k)})\},\end{aligned}$$

and $\mathbb{T}^{(k)} = \mathbb{E}^{(k)} + \frac{h_k}{3}\mathbb{M}(\mathbb{E}^{(k)})\mathbb{E}^{(k)}$ [45, 84, 85].

2. **Using 4$^{\text{th}}$-order classical Runge-Kutta method:**

$$\begin{aligned}\mathbb{E}^{(k+1)} &= \mathbb{E}^{(k)} + \frac{h_k}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\ k_1 &= \mathbb{M}(\mathbb{E}^{(k)})\mathbb{E}^{(k)}, \\ k_2 &= \mathbb{M}(\mathbb{E}^{(k)} + \frac{h_k}{2}k_1)(\mathbb{E}^{(k)} + \frac{h_k}{2}k_1), \\ k_3 &= \mathbb{M}(\mathbb{E}^{(k)} + \frac{h_k}{2}k_2)(\mathbb{E}^{(k)} + \frac{h_k}{2}k_2), \\ k_4 &= \mathbb{M}(\mathbb{E}^{(k)} + h_k k_3)(\mathbb{E}^{(k)} + h_k k_3),\end{aligned} \tag{4.4}$$

which can be rewritten as

$$\begin{aligned}\mathbb{E}^{(k+1)} &= \mathbb{E}^{(k)} + \frac{h_k}{6}\mathbb{M}(\mathbb{E}^{(k)})\mathbb{E}^{(k)} \\ &+ \{\frac{h_k}{3}\mathbb{M}(\mathbb{Z}^{(k)}) + \frac{h_k^2}{6}\mathbb{M}(\mathbb{Z}^{(k)})\mathbb{M}(\mathbb{E}^{(k)})\}\mathbb{E}^{(k)} \\ &+ \{\frac{h_k}{3}\mathbb{M}(\mathbb{V}^{(k)}) + \frac{h_k^2}{6}\mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)}) \\ &+ \frac{h_k^3}{12}\mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)})\mathbb{M}(\mathbb{E}^{(k)})\}\mathbb{E}^{(k)} \\ &+ \frac{h_k}{6}\mathbb{M}(\mathbb{E}^{(k)} + h_k\mathbb{M}(\mathbb{V}^{(k)})\mathbb{E}^{(k)} + \frac{h_k^2}{2}\mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)})\mathbb{Z}^{(k)}) \\ &\times \{\mathbb{I} + h_k\mathbb{M}(\mathbb{V}^{(k)}) + \frac{h_k^2}{2}\mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)}) \\ &+ \frac{h_k^3}{4}\mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)})\mathbb{M}(\mathbb{E}^{(k)})\}\mathbb{E}^{(k)},\end{aligned} \tag{4.5}$$

where $\mathbb{Z}^{(k)} = \mathbb{E}^{(k)} + \frac{h_k}{2}\mathbb{M}(\mathbb{E}^{(k)})\mathbb{E}^{(k)}$, and $\mathbb{V}^{(k)} = \mathbb{E}^{(k)} + \frac{h_k}{2}\mathbb{M}(\mathbb{Z}^{(k)})\mathbb{Z}^{(k)}$. The time-discrete equation is obtained as

$$\mathbb{E}^{(k+1)} = \mathbb{M}^{(k)}\mathbb{E}^{(k)}, \tag{4.6}$$

with the matrix $\mathbb{M}^{(k)}$ defined as follows:

$$\begin{aligned}\mathbb{M}^{(k)} \quad := \quad & \mathbb{I} + \frac{h_k}{6}\{\mathbb{M}(\mathbb{E}^{(k)}) + 2\mathbb{M}(\mathbb{Z}^{(k)}) + 2\mathbb{M}(\mathbb{V}^{(k)}) + \mathbb{M}(\mathbb{T}^{(k)})\} \\ & + \frac{h_k^2}{6}\{\mathbb{M}(\mathbb{Z}^{(k)})\mathbb{M}(\mathbb{E}^{(k)}) + \mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)}) + \mathbb{M}(\mathbb{T}^{(k)})\mathbb{M}(\mathbb{V}^{(k)})\} \\ & + \frac{h_k^3}{12}\{\mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)})\mathbb{M}(\mathbb{E}^{(k)}) + \mathbb{M}(\mathbb{T}^{(k)})\mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)})\} \\ & + \frac{h_k^4}{24}\{\mathbb{M}(\mathbb{T}^{(k)})\mathbb{M}(\mathbb{V}^{(k)})\mathbb{M}(\mathbb{Z}^{(k)})\mathbb{M}(\mathbb{E}^{(k)})\},\end{aligned}$$

where $\mathbb{T}^{(k)} = \mathbb{E}^{(k)} + h_k\mathbb{M}(\mathbb{V}^{(k)})\mathbb{V}^{(k)}$.

The approximate values of the next state can be obtained from the previous one by using the above iterative formulas. The DNA microarray experimental data and the environmental items obtained at the time-level t_k are represented by the vector $\bar{\mathbb{E}}^{(\kappa)}$ ($\kappa = 0, 1, \ldots, N-1$; N: the number of biological measurements) in the extended space. The approximations in the sense of (4.3) or (4.6) are denoted by $\widehat{\mathbb{E}}^{(\kappa)}$ ($\kappa = 0, 1, \ldots, N-1$), and set $\widehat{\mathbb{E}}^{(0)} = \mathbb{E}^{(0)}$. The k^{th}- approximation or prediction, $\widehat{\mathbb{E}}^{(k)}$, is calculated as $\widehat{\mathbb{E}}^{(k)}(:= \mathbb{E}^{(k)}) = \mathbb{M}^{(k-1)}(\mathbb{M}^{(k-2)} \cdots (\mathbb{M}^{(1)}(\mathbb{M}^{(0)}\mathbb{E}^{(0)})))$, where $h_k := t_{k+1} - t_k$ and $k \in \mathbb{N}_0$. We obtain our gene-environment networks by the time-discrete dynamics using formula (4.3) or (4.6). The genes and environmental items are represented by the nodes (vertices) of our network; the interactions between them are reflected by the edges, weighted with effects. The significant entry of $\mathbb{M}^{(k)}$, say, $m_{ij}^{(k)}$, is the coefficient of proportionality (i.e., multiplied by $\mathbb{E}_j^{(k)}$). It describes that the i^{th}- gene (or environmental factor) becomes changed by the j^{th}-gene (or environmental factor or the cumulative environmental item) in the step from time level k to $k+1$ [84].

4.2 Corresponding Matrix Algebra

We refer to the canonical form of matrix partitioning, given in [11, 42, 46, 130], for the time-continuous model in Eq. (3.16) and Eq. (3.17) as

$$\mathbb{M}(\mathbb{E}) = \begin{pmatrix} \mathbf{M}(\mathbf{E})_{n\times n} & \check{\mathbf{M}}(\mathbf{E})_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \end{pmatrix}, \tag{4.7}$$

where $\mathbf{M}(\mathbf{E})$ and $\check{\mathbf{M}}(\mathbf{E})$ are the matrices having dimensions $n \times n$ and $n \times m$, respectively. Herewith, the format of the matrix $\mathbb{M}(\mathbb{E})$ is $(n+m)\times(n+m)$ and $\mathbb{E} = (\mathbf{E}^T, \check{\mathbf{E}}^T)^T$ is an $(n+m)$-vector. The relations of the n genes and the m environmental factors, which describe the

structure of the gene and gene-environment network, are represented by the matrices $\mathbb{M}^{(k)}$. These matrices will be the basis of the networks. The product of two such canonical matrices is again canonical (see [46, 11, 29, 42, 43, 130] and their references).

After some notation and simplification we find that

1. **Using 3$^{\mathrm{rd}}$-order Heun's method:**

$$\begin{aligned}\mathbb{M}^{(k)} = \mathbb{I} \quad &+ \quad \frac{h_k}{4}\begin{pmatrix} \mathbf{M}(\mathbf{E}^{(k)})_{n\times n} & \check{\mathbf{M}}(\mathbf{E}^{(k)})_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m}\end{pmatrix} + \frac{3h_k}{4}\begin{pmatrix} \mathbf{A}_{n\times n} & \tilde{\mathbf{A}}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m}\end{pmatrix} \\ &+ \quad \frac{h_k^2}{2}\begin{pmatrix} \mathbf{B}_{n\times n} & \tilde{\mathbf{B}}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m}\end{pmatrix} + \frac{h_k^3}{6}\begin{pmatrix} \mathbf{C}_{n\times n} & \tilde{\mathbf{C}}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m}\end{pmatrix},\end{aligned}$$

where

$$\begin{aligned}\mathbf{A} &:= \mathbf{M}(\mathbf{E}^{(k)} + \frac{2h_k}{3}(\mathbf{M}(\mathbf{T}^{(k)})\mathbf{T}^{(k)} + \check{\mathbf{M}}(\mathbf{T}^{(k)})\check{\mathbf{T}}^{(k)})),\\
\widetilde{\mathbf{A}} &:= \check{\mathbf{M}}(\mathbf{E}^{(k)} + \frac{2h_k}{3}(\mathbf{M}(\mathbf{T}^{(k)})\mathbf{T}^{(k)} + \check{\mathbf{M}}(\mathbf{T}^{(k)})\check{\mathbf{T}}^{(k)})),\\
\mathbf{B} &:= \mathbf{M}(\mathbf{E}^{(k)} + \frac{2h_k}{3}(\mathbf{M}(\mathbf{T}^{(k)})\mathbf{T}^{(k)} + \check{\mathbf{M}}(\mathbf{T}^{(k)})\check{\mathbf{T}}^{(k)}))\mathbf{M}(\mathbf{T}^{(k)}),\\
\widetilde{\mathbf{B}} &:= \mathbf{M}(\mathbf{E}^{(k)} + \frac{2h_k}{3}(\mathbf{M}(\mathbf{T}^{(k)})\mathbf{T}^{(k)} + \check{\mathbf{M}}(\mathbf{T}^{(k)})\check{\mathbf{T}}^{(k)}))\check{\mathbf{M}}(\mathbf{T}^{(k)}),\\
\mathbf{C} &:= \mathbf{M}(\mathbf{E}^{(k)} + \frac{2h_k}{3}(\mathbf{M}(\mathbf{T}^{(k)})\mathbf{T}^{(k)} + \check{\mathbf{M}}(\mathbf{T}^{(k)})\check{\mathbf{T}}^{(k)}))\mathbf{M}(\mathbf{T}^{(k)})\mathbf{M}(\mathbf{E}^{(k)}),\\
\widetilde{\mathbf{C}} &:= \mathbf{M}(\mathbf{E}^{(k)} + \frac{2h_k}{3}(\mathbf{M}(\mathbf{T}^{(k)})\mathbf{T}^{(k)} + \check{\mathbf{M}}(\mathbf{T}^{(k)})\check{\mathbf{T}}^{(k)}))\mathbf{M}(\mathbf{T}^{(k)})\check{\mathbf{M}}(\mathbf{E}^{(k)}),\end{aligned} \tag{4.8}$$

and $\mathbb{T} := (\mathbf{T}^T, \check{\mathbf{T}}^T)^T$ is an $(n+m)$-vector with $\mathbf{T}^{(k)} := \mathbf{E}^{(k)} + \frac{h_k}{3}\{\mathbf{M}(\mathbf{E}^{(k)})\mathbf{E}^{(k)} + \check{\mathbf{M}}(\mathbf{E}^{(k)})\check{\mathbf{E}}^{(k)}\}$, $\check{\mathbf{T}}^{(k)} := \check{\mathbf{E}}^{(k)}$, and $\mathbb{I} := \mathbf{I}_d$ is a $(d \times d)$-unit matrix with $d = n + m$ [45, 84, 85].

2. **Using 4$^{\mathrm{th}}$-order classical Runge-Kutta method:**

$$\begin{aligned}\mathbb{M}^{(k)} = \mathbb{I} \quad &+ \quad \frac{h_k}{6}\begin{pmatrix} \mathbf{A}_{n\times n} & \tilde{\mathbf{A}}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m}\end{pmatrix} + \frac{h_k^2}{6}\begin{pmatrix} \mathbf{B}_{n\times n} & \tilde{\mathbf{B}}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m}\end{pmatrix} \\ &+ \quad \frac{h_k^3}{12}\begin{pmatrix} \mathbf{C}_{n\times n} & \tilde{\mathbf{C}}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m}\end{pmatrix} + \frac{h_k^4}{24}\begin{pmatrix} \mathbf{D}_{n\times n} & \tilde{\mathbf{D}}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m}\end{pmatrix},\end{aligned}$$

with

$$
\begin{aligned}
\mathbf{A} &:= \mathbf{M}(\mathbf{E}^{(k)}) + 2\mathbf{M}(\mathbf{Z}^{(k)}) + 2\mathbf{M}(\mathbf{V}^{(k)}) + \mathbf{M}(\mathbf{T}^{(k)}),\\
\widetilde{\mathbf{A}} &:= \check{\mathbf{M}}(\mathbf{E}^{(k)}) + 2\check{\mathbf{M}}(\mathbf{Z}^{(k)}) + 2\check{\mathbf{M}}(\mathbf{V}^{(k)}) + \check{\mathbf{M}}(\mathbf{T}^{(k)}),\\
\mathbf{B} &:= \mathbf{M}(\mathbf{Z}^{(k)})\mathbf{M}(\mathbf{E}^{(k)}) + \mathbf{M}(\mathbf{V}^{(k)})\mathbf{M}(\mathbf{Z}^{(k)}) + \mathbf{M}(\mathbf{T}^{(k)})\mathbf{M}(\mathbf{V}^{(k)}),\\
\widetilde{\mathbf{B}} &:= \mathbf{M}(\mathbf{Z}^{(k)})\check{\mathbf{M}}(\mathbf{E}^{(k)}) + \mathbf{M}(\mathbf{V}^{(k)})\check{\mathbf{M}}(\mathbf{Z}^{(k)}) + \mathbf{M}(\mathbf{T}^{(k)})\check{\mathbf{M}}(\mathbf{V}^{(k)}),\\
\mathbf{C} &:= \mathbf{M}(\mathbf{V}^{(k)})\mathbf{M}(\mathbf{Z}^{(k)})\mathbf{M}(\mathbf{E}^{(k)}) + \mathbf{M}(\mathbf{T}^{(k)})\mathbf{M}(\mathbf{V}^{(k)})\mathbf{M}(\mathbf{Z}^{(k)}),\\
\widetilde{\mathbf{C}} &:= \mathbf{M}(\mathbf{V}^{(k)})\mathbf{M}(\mathbf{Z}^{(k)})\check{\mathbf{M}}(\mathbf{E}^{(k)}) + \mathbf{M}(\mathbf{T}^{(k)})\mathbf{M}(\mathbf{V}^{(k)})\check{\mathbf{M}}(\mathbf{Z}^{(k)}),\\
\mathbf{D} &:= \mathbf{M}(\mathbf{T}^{(k)})\mathbf{M}(\mathbf{V}^{(k)})\mathbf{M}(\mathbf{Z}^{(k)})\mathbf{M}(\mathbf{E}^{(k)}),\\
\widetilde{\mathbf{D}} &:= \mathbf{M}(\mathbf{T}^{(k)})\mathbf{M}(\mathbf{V}^{(k)})\mathbf{M}(\mathbf{Z}^{(k)})\check{\mathbf{M}}(\mathbf{E}^{(k)}),
\end{aligned}
\tag{4.9}
$$

where

$\mathbf{Z}^{(k)} := \mathbf{E}^{(k)} + \frac{h_k}{2}\{\mathbf{M}(\mathbf{E}^{(k)})\mathbf{E}^{(k)} + \check{\mathbf{M}}(\mathbf{E}^{(k)})\check{\mathbf{E}}^{(k)}\}$,
$\mathbf{V}^{(k)} := \mathbf{E}^{(k)} + \frac{h_k}{2}\{\mathbf{M}(\mathbf{Z}^{(k)})\mathbf{Z}^{(k)} + \check{\mathbf{M}}(\mathbf{Z}^{(k)})\check{\mathbf{Z}}^{(k)}\}$,
$\mathbf{T}^{(k)} := \mathbf{E}^{(k)} + h_k\{\mathbf{M}(\mathbf{V}^{(k)})\mathbf{V}^{(k)} + \check{\mathbf{M}}(\mathbf{V}^{(k)})\check{\mathbf{V}}^{(k)}\}$,
$\check{\mathbf{Z}}^{(k)} = \check{\mathbf{V}}^{(k)} = \check{\mathbf{T}}^{(k)} := \check{\mathbf{E}}^{(k)}$.

Note that, $\mathbb{Z} := (\mathbf{Z}^T, \check{\mathbf{Z}}^T)^T$, $\mathbb{V} := (\mathbf{V}^T, \check{\mathbf{V}}^T)^T$, $\mathbb{T} := (\mathbf{T}^T, \check{\mathbf{T}}^T)^T$ are $(n+m)$-vectors all and $\mathbb{I} := \mathbf{I}_d$ is the $(d \times d)$-unit matrix with $d = n + m$ [84].

Therefore, $\mathbb{M}^{(k)}$ in (4.8) and (4.9) has its final canonical block form as:

$$
\begin{pmatrix}
\widetilde{\mathbf{M}(\mathbf{E}^{(k)})}_{n\times n} & \widetilde{\check{\mathbf{M}}(\mathbf{E}^{(k)})}_{n\times m}\\
\mathbf{0}_{m\times n} & \mathbf{I}_{m\times m}
\end{pmatrix}.
\tag{4.10}
$$

4.3 Numerical Applications and Comparisons

This part of the thesis contains the major part of the original results of the numerical study for the improvement in the representation and prediction of time-discrete dynamics of gene-environment (also target-environment) regulatory networks. Two significant examples are investigated in details by using with two different types of data sets.

4.3.1 Example with an Artificial Data Set

In this subsection of the thesis, an example with an artificial data set is studied in details in order to apply our newly derived numerical schemes and compare them with the previously studied schemes in the literature. In this respect, we examine their performance and behavior firstly by applying all of the four schemes (as *a class of explicit Runge-Kutta methods*), Euler method, 2^{nd}-order Heun's method, 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method, for a fixed gene-expression data set and its approximated derivative data with a fixed step-size. The obtained results are compared and represented in graphs. Then, we choose different step-sizes in the implementation of all these four methods and compare the rate of convergence. Finally, the given gene expression data is perturbed by different choices of the perturbation value ϵ and the resulting changes in the behavior of the genes are watched in order to detect the allowable interval for perturbations.

4.3.1.1 Studied Model

The gene network model that we discuss and consider in our examples is represented by the differential equation

$$\dot{\mathbf{E}} = \mathbf{M}(\mathbf{E})\mathbf{E}, \tag{4.11}$$

that is described in Chapter 3, Section 3.1, and where $\mathbf{M}$ is a constant $(n \times n)$-matrix.

Our *aim* is to compute a gene network (represented by the matrix $\mathbf{M}$) based on gene expression data. In order to do this, we solve a MINLP problem that is defined in the following way.

The objective function of our MINLP problem is the following:

$$\min_{\mathbf{M}=(m_{ij})} \quad \sum_{k=1}^{N} \left\| \mathbf{M}\bar{\mathbf{E}}^{(k)} - \dot{\bar{\mathbf{E}}}^{(k)} \right\|_2^2, \tag{4.12}$$

that means, we want to find a matrix $\mathbf{M}$ such that the distances between the forecasted and the actual observed values are as small as possible with respect to the $\|\cdot\|_2$ norm. Here, N is the number of biological measurements and the $\dot{\bar{\mathbf{E}}}^{(k)}$ are the difference quotients based on the k^{th}-experimental data $\bar{\mathbf{E}}^{(k)}$ with step lengths h_k between neighbouring sampling times [10, 28, 36, 45, 46, 84, 85].

Because of a high degree of freedom in the problem, it is needed to restrict the solution space according to the underlying biological motivation [28, 36, 37] and mathematical reasoning that are explained in Section 3.3 in previous chapter. Otherwise, a very big amount of expression data is necessary to solve the minimization problem in (4.12). The values m_{ij} for $i \neq j$ are nonnegative, since no gene consumes another one, and $m_{ij} = 0$ means that the two genes i and j do not interact at all. A constant positive vector $\boldsymbol{\lambda} \in \mathbb{R}^n$ represents the lower bound for the amount of decrease of the transcript concentration between two time steps which is referred as *degradation rate* [28, 36, 37]. Therefore, for $i, j \in G$ (where $G = \{1, 2, \ldots, n\}$ is the set of genes but environmental factors could be included here, too) we have

$$m_{ij} \geq \begin{cases} -\lambda(i), & i = j, \\ 0, & i \neq j. \end{cases} \tag{4.13}$$

By the above condition, a bound is imposed on self-degradation of the genes by the term $-\boldsymbol{\lambda}$ and off-diagonal genes are prevented from negative regulation. This situation is a very special case and, here, we can suppose to consider some further cases like removing the nonnegativity condition as

$$m_{ij} \geq -\lambda(i), \quad i = j, \tag{4.14}$$

or adding parameters for off-diagonal elements by

$$m_{ij} \geq \begin{cases} -\lambda(i), & i = j, \\ -\delta(i, j), & i \neq j. \end{cases} \tag{4.15}$$

where the positive vectors of parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ are determined with the colleagues from biology, medicine and from environmental sciences. All of these bounds can be regarded as realizations of the case whose solutions are included in efficiency frontiers (in the case of parametric variation of one bound) or efficiency surface (in the case of parametric variation of several bounds) as mentioned in Section 3.3.

We want to emphasize that the first extended case in (4.14) is studied with a given data in Section 4.3.2 and obtained results are presented in Subsubsection 4.3.2.3 with comparisons.

To obtain a relatively sparse network, it is needed to limit the maximum outdegree and indegree of each node. In order not to lose the decomposition property of the minimization problem by limiting the maximum outdegree, we bound the *indegree* of each gene i by a given parameter $deg_{max,i} \in \mathbb{N}_0$. So, in order to bound the indegree of each node, we introduce binary variables $y_{ij} \in \{0, 1\}$ in the subsequent way [28, 36, 37]:

$$y_{ij} = \begin{cases} 0, & \text{if } m_{ij} = 0, \\ 1, & \text{if } m_{ij} \neq 0. \end{cases} \tag{4.16}$$

We formulate (4.16) as the following nonlinear constraints for our model:

$$(1 - y_{ij}) \cdot m_{ij} = 0, \quad \forall\, i, j \in G. \tag{4.17}$$

Now, the number of nonzero entries per row of the matrix $\mathbf{M} = (m_{ij})_{1 \leq i,j \leq n}$ can be limited by the degree number, $deg_{max,i}$, which is content of the following constraints:

$$\sum_{j \in G} y_{ij} \leq deg_{max,i}\,, \quad \forall\, i \in G. \tag{4.18}$$

After considering all these constraints, we aim to solve the MINLP problem

$$\textit{(OP-I)} \qquad \min \quad (4.12), \qquad \text{s.t.} \quad \{(4.13), (4.17), (4.18)\}, \tag{4.19}$$

to proven global optimality. Note that we recall the above MINLP problem as our *original* optimization problem and denote it as *(OP-I)*. An *extended* version of it, called *(OP-II)* is also defined and studied in the following Subsection 4.3.2 concerning the numerical example with real-world data.

4.3.1.2 Comparison of Methods for Fixed Step-Size

Here, we numerically solve the problem in (4.19), called *(OP-I)*, within the model described by (4.11). We have four different genes and their expression levels at four different times according to the following Table 4.1 (from [36]).

We use an equally-spaced time discretization as $h_k = 1\ \forall k = 1, 2, ..., N - 1$. In [45, 84, 85], we apply the 3^{rd}-order Heun's method to approximate the $\dot{\mathbf{E}}(t)$ according to the above given data and obtain the following approximate derivative values

$$\begin{aligned} \dot{\mathbf{E}}_1^T &= [0, \quad -50, \quad 50, \quad -255], \\ \dot{\mathbf{E}}_2^T &= [0, \quad -20, \quad 20, \quad 255], \\ \dot{\mathbf{E}}_3^T &= [0, \quad -20, \quad 20, \quad -255], \end{aligned} \tag{4.20}$$

Table 4.1: Expression scores of the genes A, B, C and D at four time points

Time / Genes	A	B	C	D	
1	255	250	0	255	$= \bar{\mathbf{E}}_1^T$
2	255	200	50	0	$= \bar{\mathbf{E}}_2^T$
3	255	180	70	255	$= \bar{\mathbf{E}}_3^T$
4	255	170	80	0	$= \bar{\mathbf{E}}_4^T$

where $\dot{\bar{\mathbf{E}}}_k$ refers to the k^{th}- approximation for the derivative $\dot{\bar{\mathbf{E}}}$ evaluated at the time point t_k, namely $\dot{\bar{\mathbf{E}}}(t_k)$ (k=1,2,3). The constraints in the mixed-integer problem in (4.19), are given biologically as [36]

$$\lambda(i) = 2, \quad deg_{max,i} = 2, \quad i = 1, 2, 3, 4. \tag{4.21}$$

Used software: The above model was formulated using the modeling language Zimpl 3.0 [142], and solved by SCIP 1.2 [143, 144] as a branch-and-cut-framework, together with SOPLEX 1.4.1 as our LP-solver [145]. Our problem was solved to proven optimality after 0.12 seconds. To this end, 28 branch-and-bound nodes had to be evaluated.

Solving the minimization problem in (4.19), we compute the following network matrix $\mathbf{M}$:

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.26 & -0.46 & 0 & 0 \\ 0.19 & 0 & -0.46 & 0 \\ 1 & 0 & 0 & -2 \end{pmatrix}, \tag{4.22}$$

where the objective function value of matrix $\mathbf{M}$ for (4.12) is 92.31. The determinant of obtained matrix $\mathbf{M}$ is zero and the eigenvalues are $\sigma_1 = -2$, $\sigma_{2,3} = -0.46$, $\sigma_4 = 0$.

Next, the 3^{rd} order Heun's time discretization formula for our model in (4.11) is derived as follows

$$\mathbf{E}_{k+1} = (\mathbf{I} + h_k\mathbf{M} + \frac{h_k^2}{2}\mathbf{M}^2 + \frac{h_k^3}{6}\mathbf{M}^3)\mathbf{E}_k, \tag{4.23}$$

where $\mathbf{I}$ is a (4×4)-identity matrix, $\mathbf{M}$ is a (4×4)-matrix given in (4.22) and $\mathbf{E}_k, \mathbf{E}_{k+1}$ are (4×1)-vectors. Lastly, by using the obtained matrix $\mathbf{M}$ and the iteration formula in Eq. (4.23), we get the approximate values of gene expressions in Table 4.2 given below [84, 85]:

Table 4.2: Approximation and extrapolation of gene expressions

Time / Genes	A	B	C	D
1	255	250	0	255
2	255	211.0011	38.9984	85
3	255	186.4872	63.5111	141.6667
4	255	171.0782	78.9187	122.7778
5	255	161.3924	88.6032	129.0741
6	255	155.3041	94.6904	126.9753
7	255	151.4771	98.5166	127.6749
8	255	149.0715	100.9216	127.4417
⋮	⋮	⋮	⋮	⋮
23	255	145.0043	104.9874	127.50
⋮	⋮	⋮	⋮	⋮
33	255	145.0004	104.9912	127.50
⋮	⋮	⋮	⋮	⋮
100	255	145.0004	104.9912	127.50

According to the generated time series in Table 4.2, we can say that the structural behavior of the obtained results is almost the same (constant first column, decreasing second column and increasing third column) with the given data in Table 4.1. For the values presented in the last column of Table 4.2, instead of an alternating behavior, we obtain a damped oscillatory behavior by using the 3^{rd}-order Heun's discretization scheme. The results for the last column converges to the mean value of 0 and 255 which has very important effect in order to reach the equilibrium point of the system.

Therefore, the approximate gene-expression time-series results obtained by 3^{rd}-order Heun's method have more smooth behavior and converging to the limit point

$$\mathbf{E}^* = [255, \quad 145.0004, \quad 104.9912, \quad 127.5]^T,$$

which is the *fixed point* (*equilibrium point*) of the considered dynamical equation $\dot{\mathbf{E}} = \mathbf{ME}$ since it satisfies the Definition 3.2.1 and so Definition 3.0.8 with the matrix calculated $\mathbf{M}$ in (4.22). Here, $\mathbf{ME}^* = [0, \quad -0.4001840, \quad 0.15404799, \quad 0]^T \simeq [0, \quad 0, \quad 0, \quad 0]^T$.

As *biological interpretation* of the obtained results for Gene D which shows an alternating behavior experimentally in the first four time levels, we can say that, fading oscillating gene expression, obtained by using 3^{rd}-order Heun's discretization, can be observed in biological systems. One well-known example is the damped oscillation of circadian rhythms (molecular clock) of CO_2 output in the leaves of plants after the trigger (day-light) has been removed [146].

We present here Figure 4.1 [45, 84, 85], in order to compare our output coming from both Euler's and 3^{rd}-order Heun's methods using the calculated matrix $\mathbf{M}$ in (4.22). It is seen that the results of 3^{rd}-order Heun's method are convergent and we reach the stable values after a few time steps.

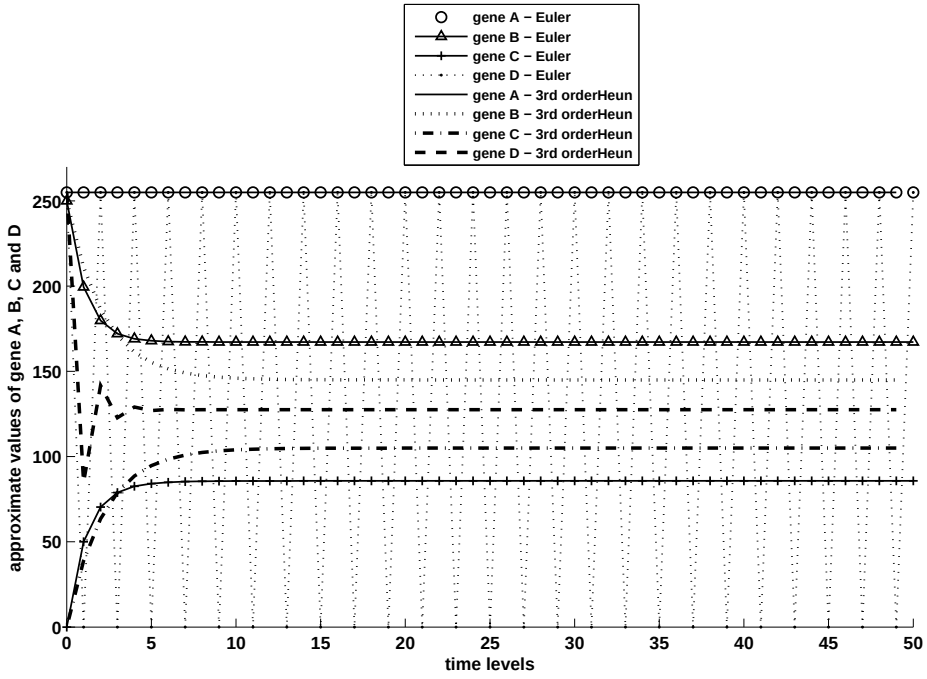


Figure 4.1: Approximate results of gene-expressions of all genes by using Euler's and 3^{rd}-order Heun's methods.

As a further step, we calculated the approximate results of gene-expressions by using four discretization methods as a class of explicit Runge-Kutta methods with the same artificial data given in Table 4.1, then compare the obtained results among them. Therefore, we apply the same procedure described above for Euler's method, also for 2^{nd}-order Heun's method, 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method for the following fixed data of $\dot{\mathbf{E}}$ (which is obtained by forward difference approximation):

$$\begin{aligned} \dot{\mathbf{E}}_1^T &= [0, \quad -50, \quad 50, \quad -255], \\ \dot{\mathbf{E}}_2^T &= [0, \quad -20, \quad 20, \quad 255], \\ \dot{\mathbf{E}}_3^T &= [0, \quad -10, \quad 10, \quad -255], \end{aligned} \tag{4.24}$$

obtained from the data in Table 4.1 and for the correspondingly calculated matrix $\mathbf{M}$ (with objective function value 2.564) given in below:

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -0.20 & 0.38 & 0 \\ 0.19 & 0 & -0.58 & 0 \\ 1 & 0 & 0 & -2 \end{pmatrix}, \tag{4.25}$$

which have the determinant zero and eigenvalues $\sigma_1 = -2$, $\sigma_2 = -0.20$, $\sigma_3 = -0.58$ and $\sigma_4 = 0$.

Then, following iterative formulas are applied for the considered model $\dot{\mathbf{E}} = \mathbf{ME}$ in order to generate approximate time series results:

$$\begin{aligned} \text{Euler's method:} \quad & \mathbf{E}^{(k+1)} = (\mathbf{I} + h_k\mathbf{M})\mathbf{E}^{(k)}, \\ 2^{nd} \text{ order Heun's method:} \quad & \mathbf{E}^{(k+1)} = (\mathbf{I} + h_k\mathbf{M} + \frac{h_k^2}{2}\mathbf{M}^2)\mathbf{E}^{(k)}, \\ 3^{rd} \text{ order Heun's method:} \quad & \mathbf{E}^{(k+1)} = (\mathbf{I} + h_k\mathbf{M} + \frac{h_k^2}{2}\mathbf{M}^2 + \frac{h_k^3}{6}\mathbf{M}^3)\mathbf{E}^{(k)}, \\ 4^{th} \text{ order classical Runge-Kutta method:} \quad & \mathbf{E}^{(k+1)} = (\mathbf{I} + h_k\mathbf{M} + \frac{h_k^2}{2}\mathbf{M}^2 + \frac{h_k^3}{6}\mathbf{M}^3 + \frac{h_k^4}{24}\mathbf{M}^4)\mathbf{E}^{(k)}. \end{aligned} \tag{4.26}$$

In the following graphs, given in Figures 4.2-4.5, we present the obtained time series results for the gene-expression values that we get from applying all these four different discretization schemes given in (4.26), for the considered fixed data in Table 4.1 and derivative data in (4.24) with step-size $h_k = 1$.

It is seen from the calculated gene-expression results by using higher degree numerical methods have smooth behavior especially for the alternating gene, Gene D, and also they are

converging to the limit point

$$\mathbf{E}^* = [255, \quad 163.779131, \quad 86.222891, \quad 127.5]^T.$$

which gives $\mathbf{ME}^* = [0, \quad 0.008872380, \quad -1.55927678, \quad 0]^T$. For the third component of the vector there is a little difference from zero vector in order for $\mathbf{E}^*$ to be the fixed point. The reason can be the choice of approximation in the derivative, $\dot{\mathbf{E}}$. Since we take the derivative data in (4.24) which is approximated by a first order approximation (forward difference) instead of a third order approximation like in (4.20).

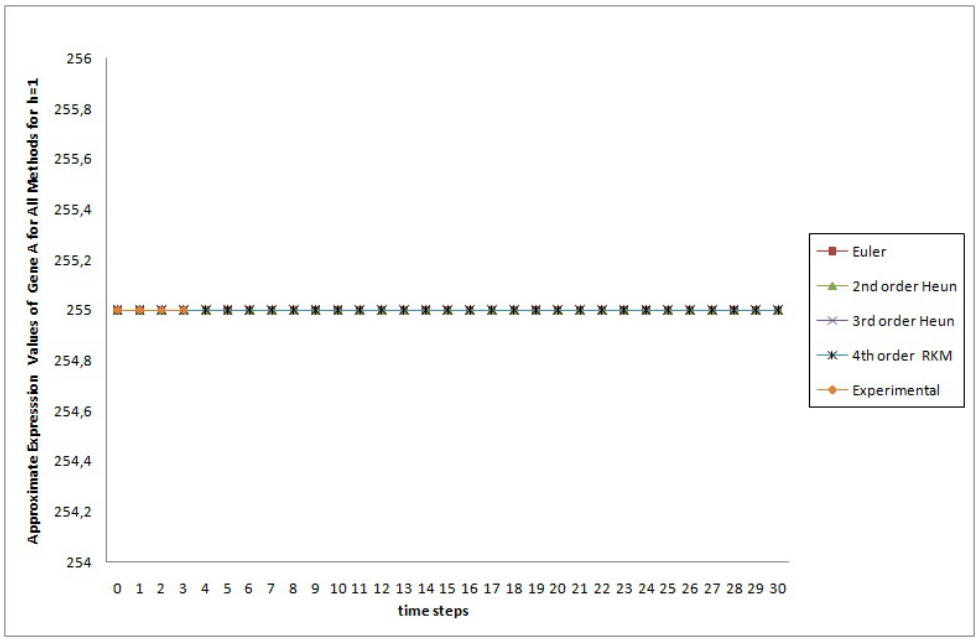


Figure 4.2: Results of Gene A using different methods for fixed data and fixed step-size

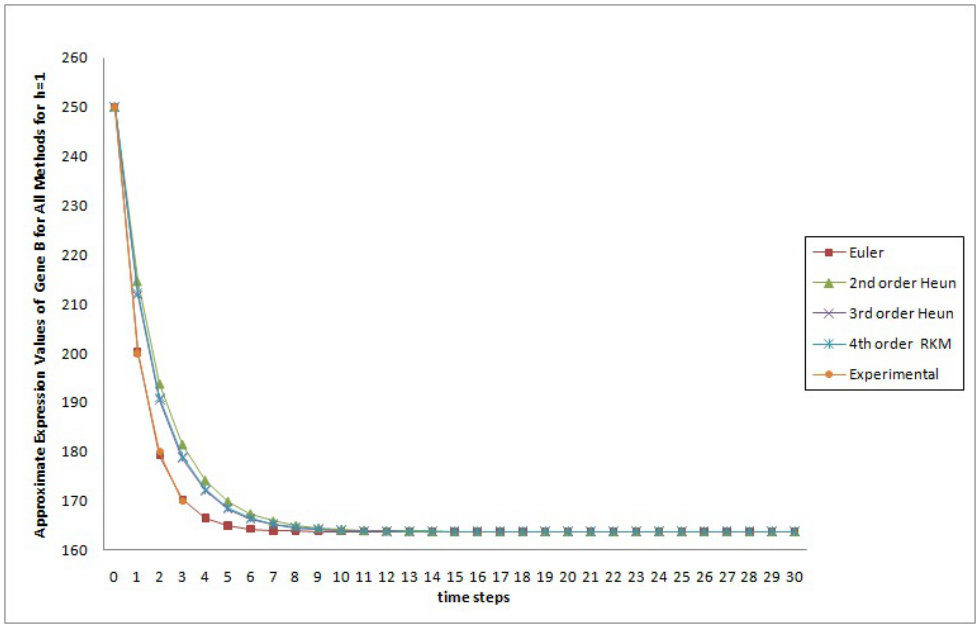


Figure 4.3: Results of Gene B using different methods for fixed data and fixed step-size

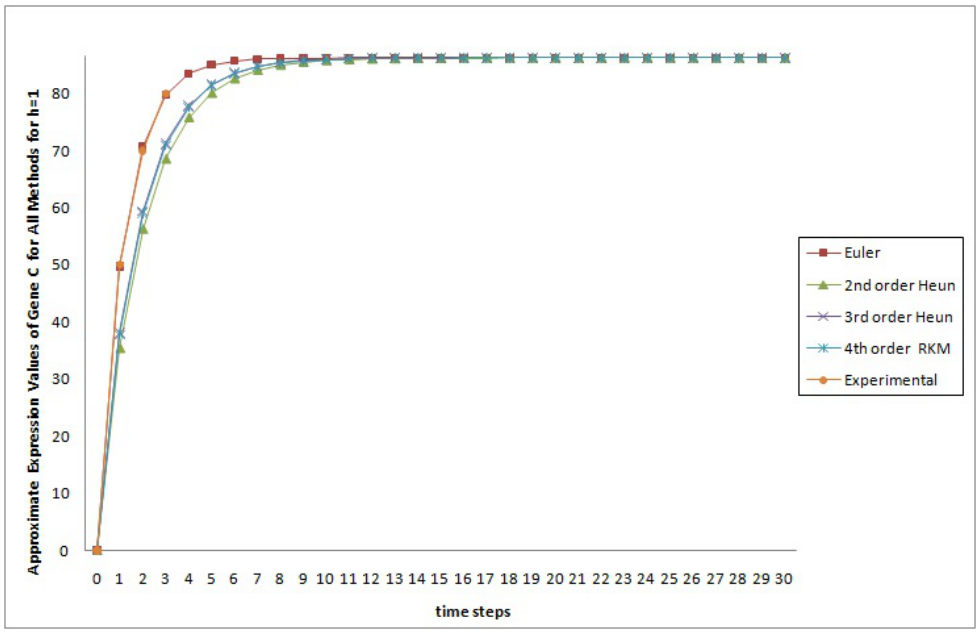


Figure 4.4: Results of Gene C using different methods for fixed data and fixed step-size

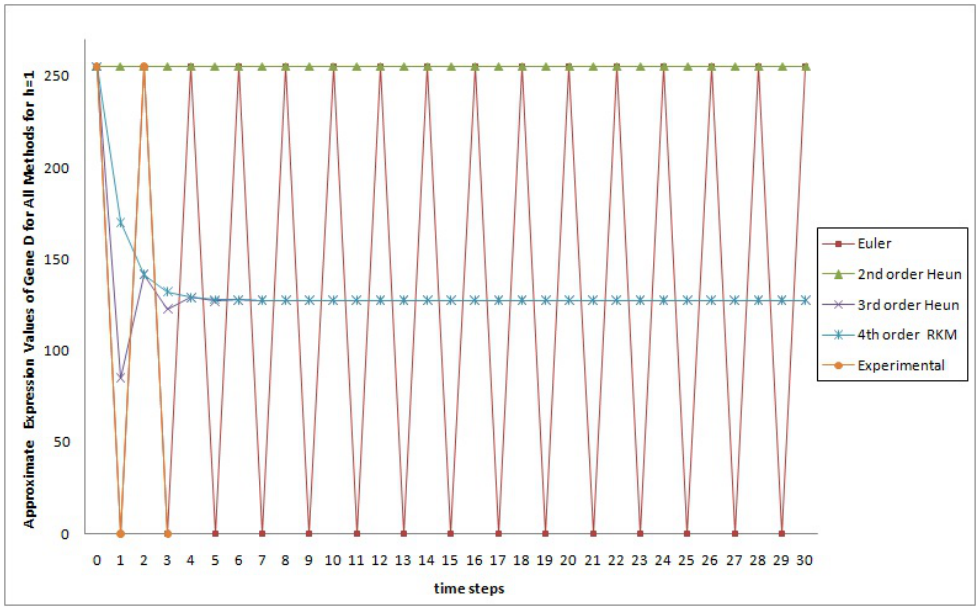


Figure 4.5: Results of Gene D using different methods for fixed data and fixed step-size

4.3.1.3 Different Step-Size Analysis

In this part of our numerical application, we study the same MINLP model in (4.19) that is called *(OP-I)* for the same artificial data in Table 4.1 and its approximated derivative data in (4.24). This time we apply all mentioned four numerical schemes for various step-sizes $h_k = 2^0, 2^{-1}, 2^{-2}, 2^{-3}$. Our aim is to see the effect of reducing the step-size in each method, gene-wise, to the produced approximate gene-expression results.

Here, we have the calculated network matrix $\mathbf{M}$ in (4.25), and then we apply each numerical scheme listed in (4.26) for all values of $h_k = 2^0, 2^{-1}, 2^{-2}, 2^{-3}$. The calculated gene-expression time-series results are presented gene by gene in the following graphs.

Note that, 50 iterations are performed for $h_k = 2^0$, 100 iterations are performed for $h_k = 2^{-1}$, 200 iterations are performed for $h_k = 2^{-2}$ and 400 iterations are performed for $h_k = 2^{-3}$ in each scheme. Then, obtained results for each scheme are collected at the same time level in terms of hours (hr) in order to do the comparison so that the corresponding figures are plotted in a meaningful way.

(i) Different step-size analysis with Euler's method

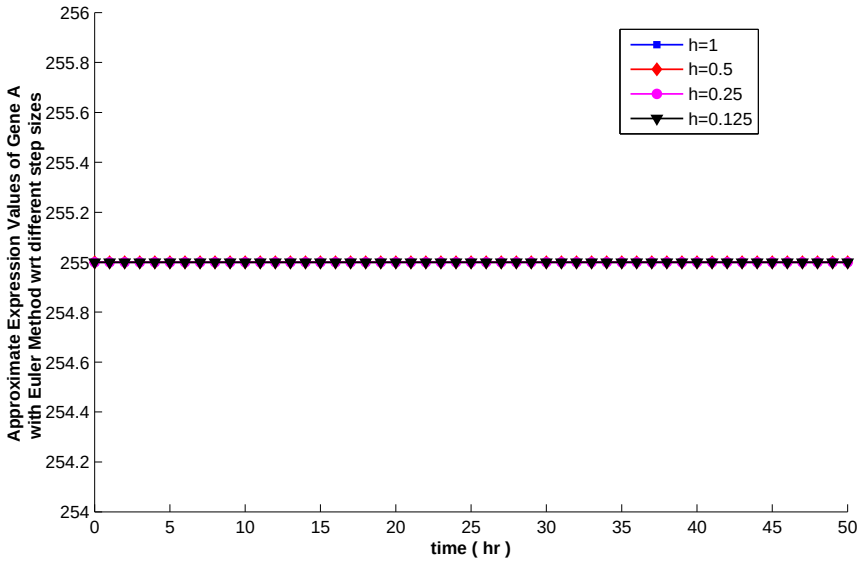


Figure 4.6: Results of Gene A using different step-sizes with Euler method

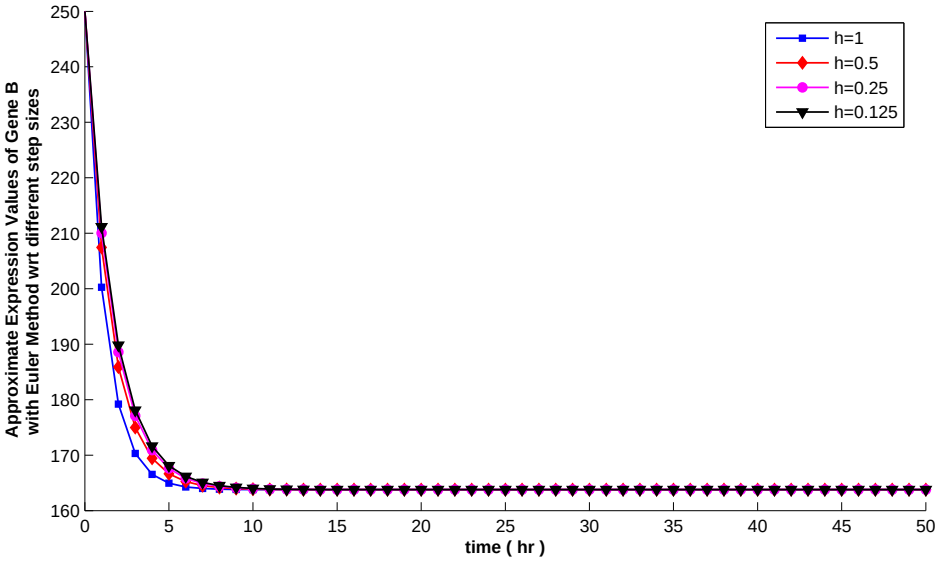


Figure 4.7: Results of Gene B using different step-sizes with Euler method

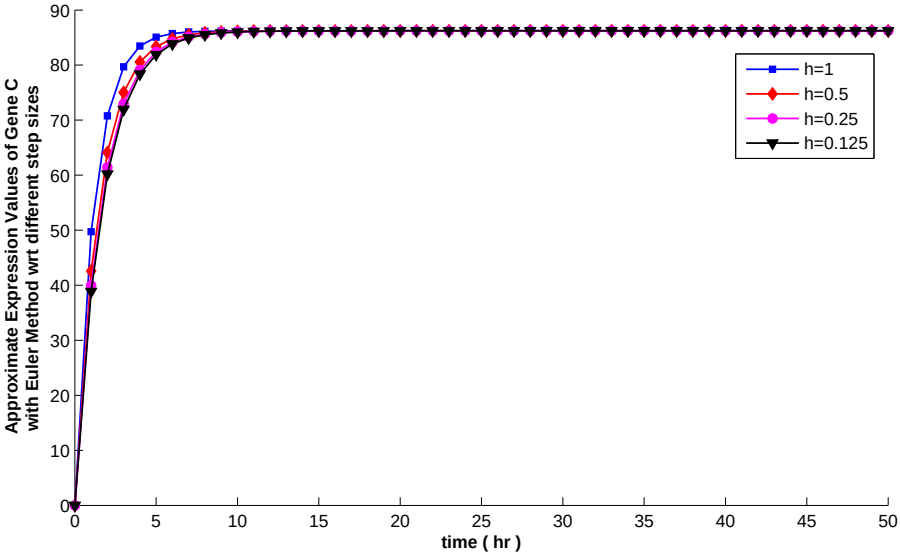


Figure 4.8: Results of Gene C using different step-sizes with Euler method

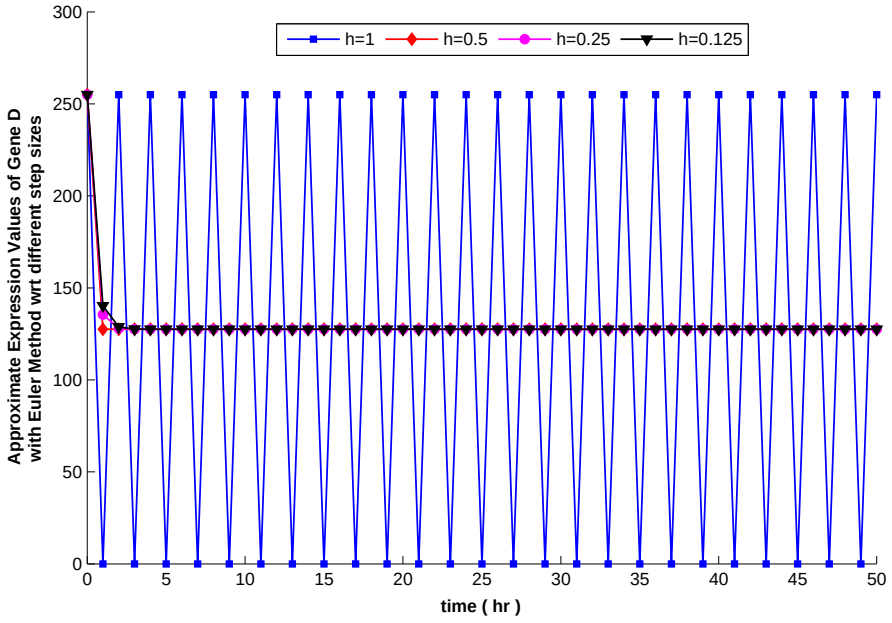


Figure 4.9: Results of Gene D using different step-sizes with Euler method

(ii) Different step-size analysis with 2^{nd}-order Heun's method

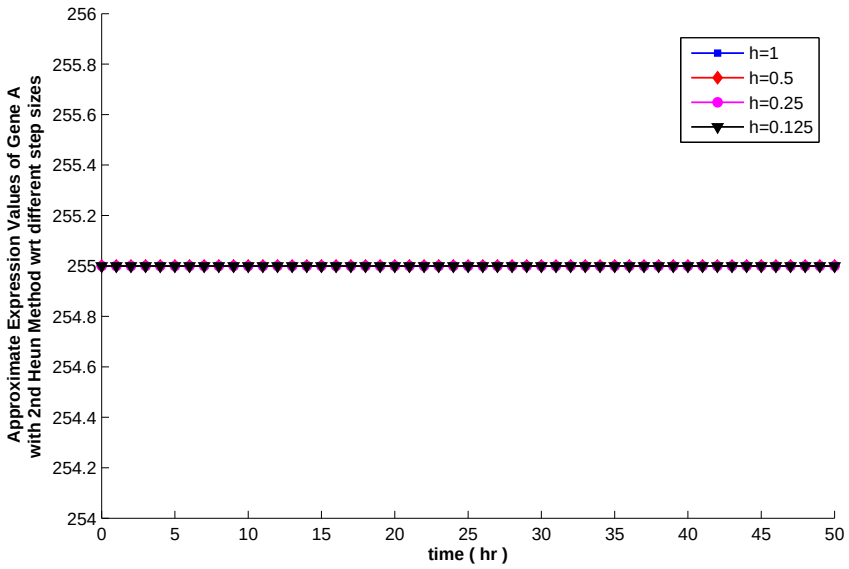


Figure 4.10: Results of Gene A using different step-sizes with 2^{nd}-order Heun's method

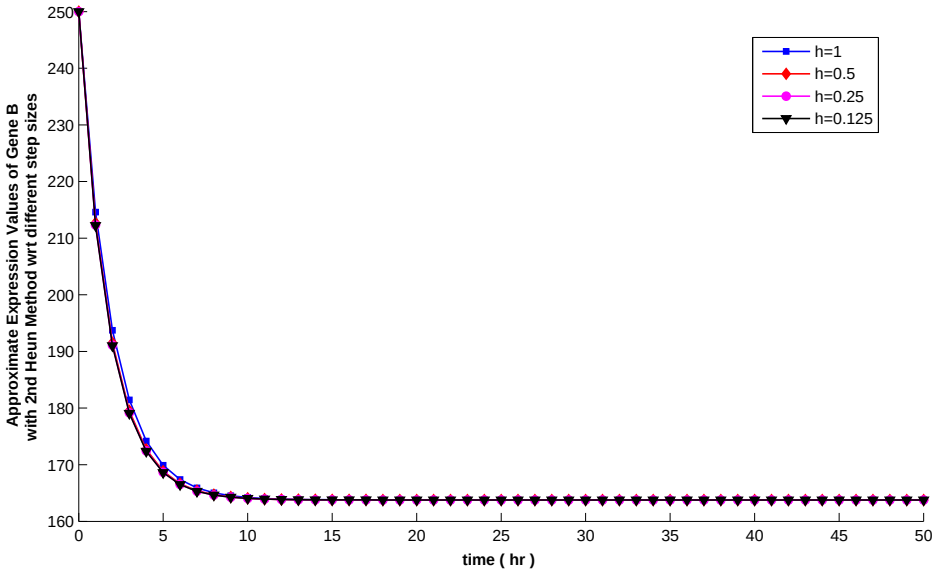


Figure 4.11: Results of Gene B using different step-sizes with 2^{nd}-order Heun's method

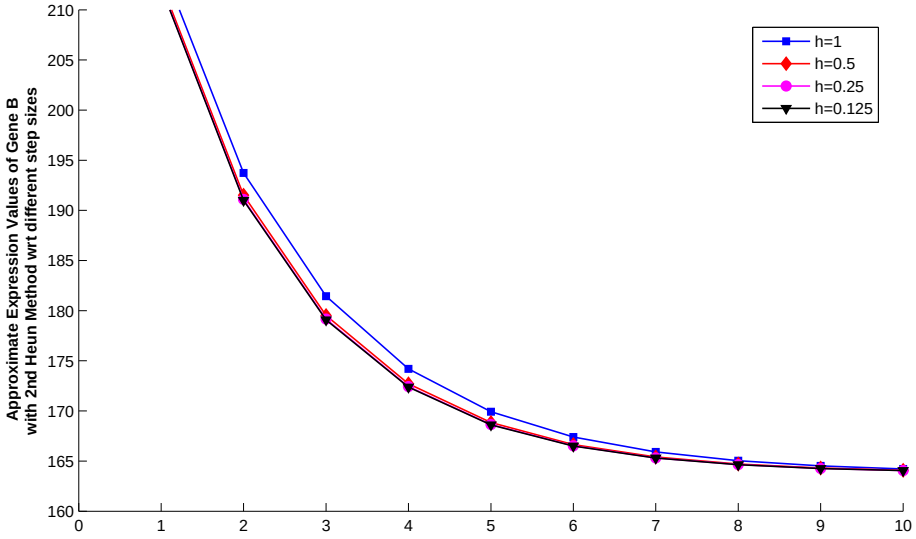


Figure 4.12: Results of Gene B with a *focused view*

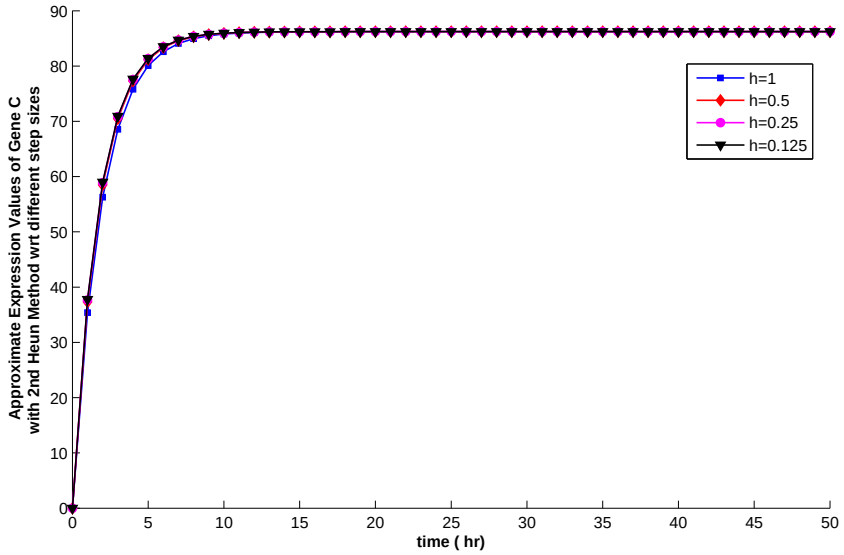


Figure 4.13: Results of Gene C using different step-sizes with 2^{nd}-order Heun's method

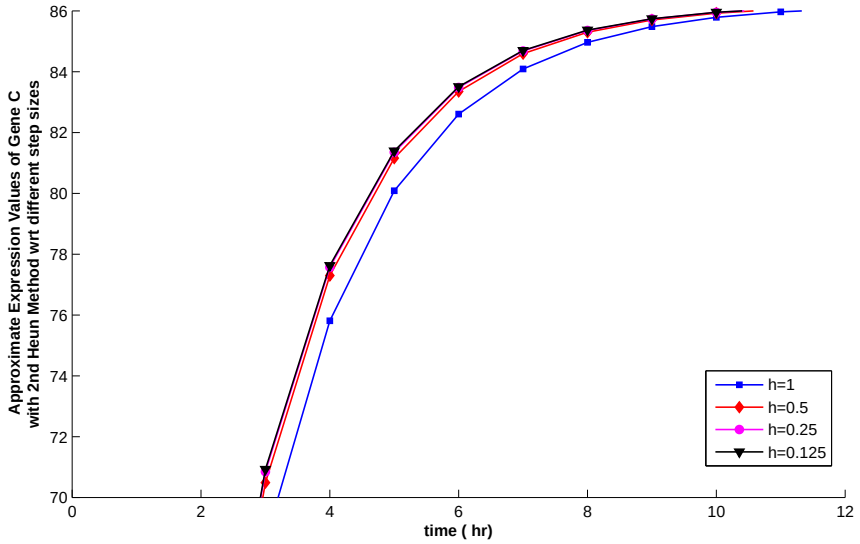


Figure 4.14: Results of Gene C with a *focused view*

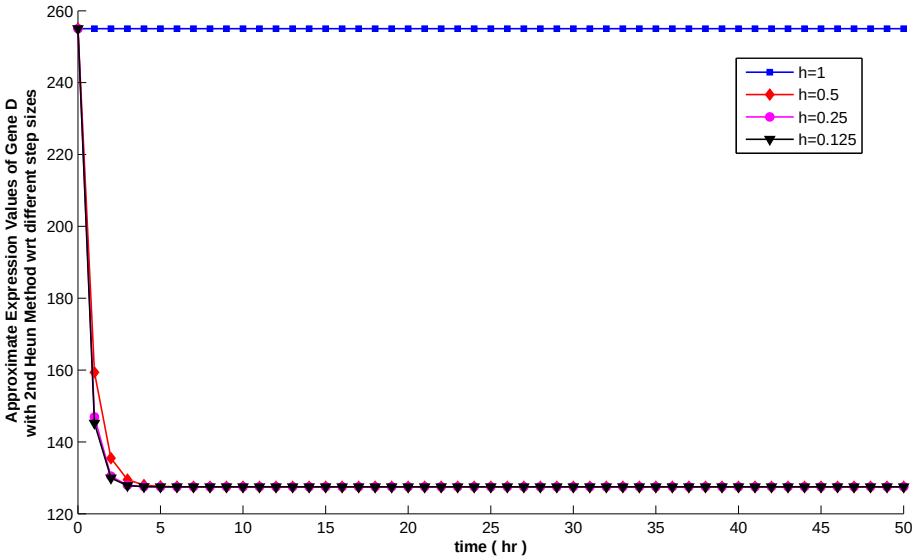


Figure 4.15: Results of Gene D using different step-sizes with 2^{nd}-order Heun's method

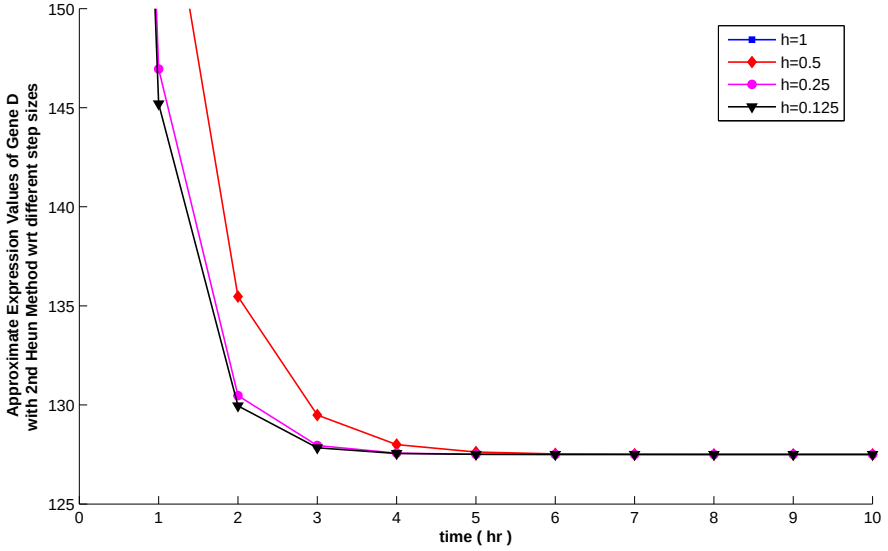


Figure 4.16: Results of Gene D with a *focused view*

(iii) Different step-size analysis with 3^{rd}-order Heun's method

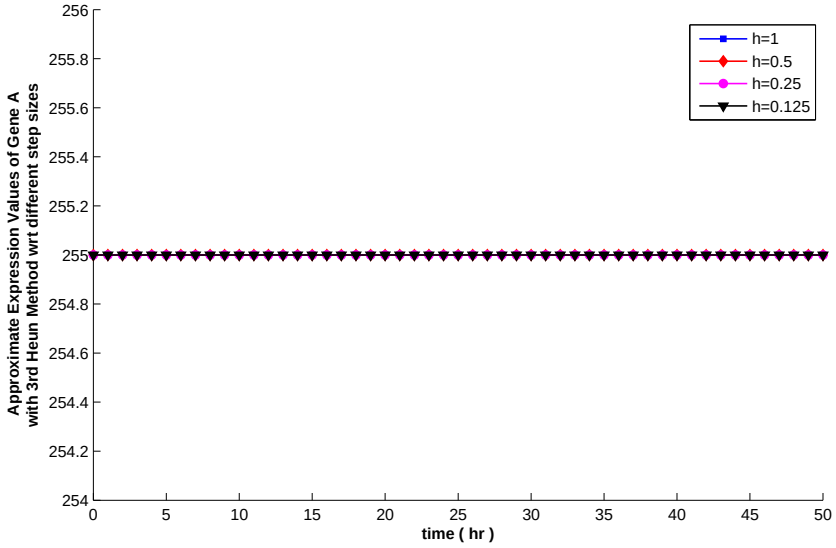


Figure 4.17: Results of Gene A using different step-sizes with 3^{rd}-order Heun's method

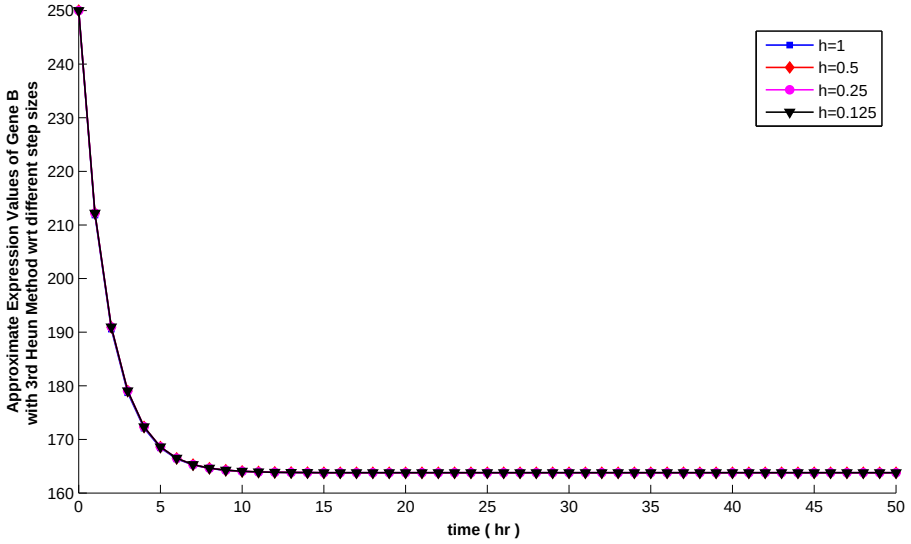


Figure 4.18: Results of Gene B using different step-sizes with 3^{rd}-order Heun's method

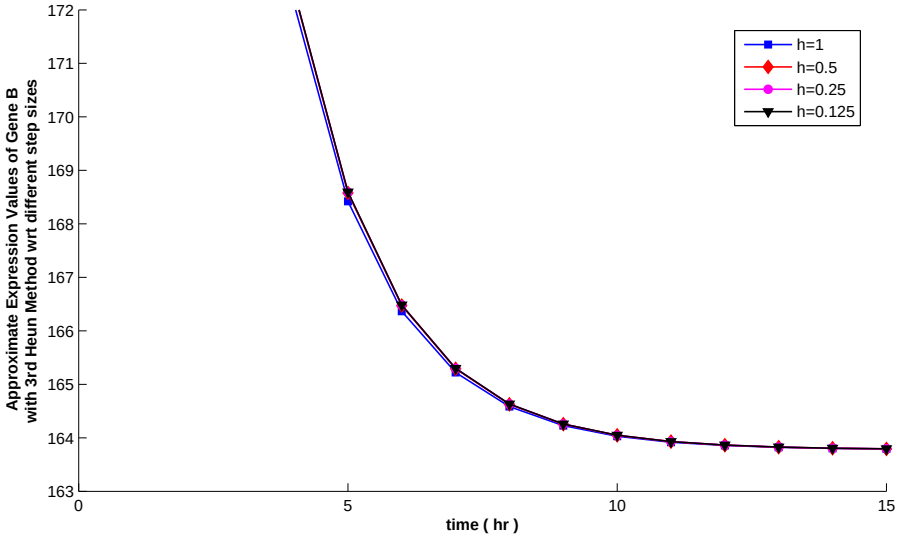


Figure 4.19: Results of Gene B with a *focused view*

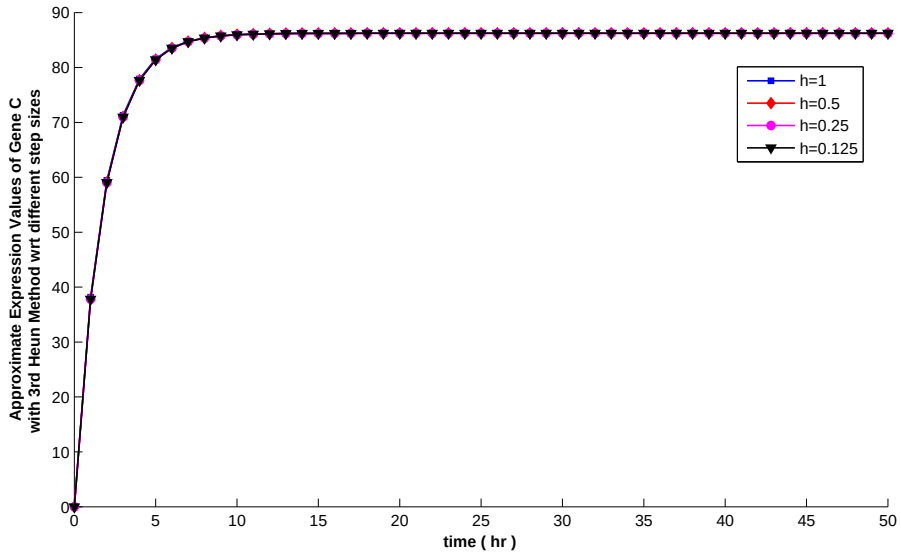


Figure 4.20: Results of Gene C using different step-sizes with 3^{rd}-order Heun's method

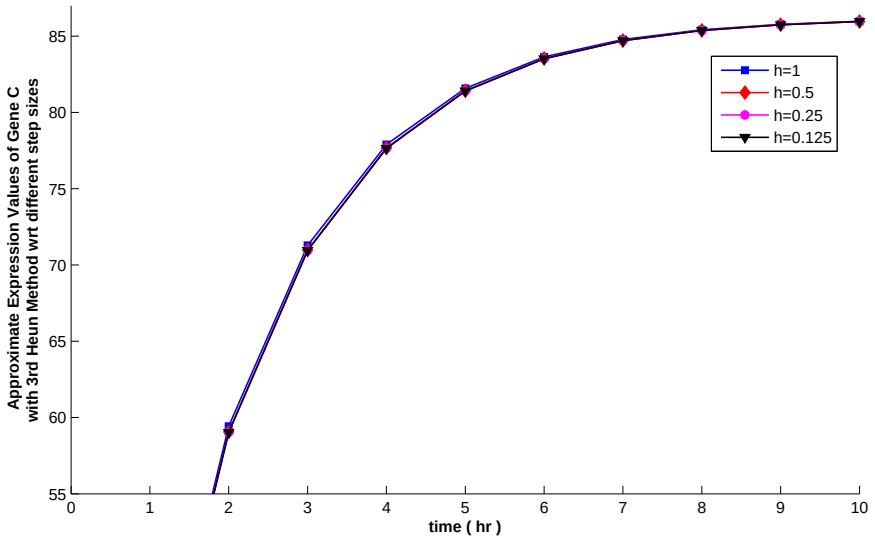


Figure 4.21: Results of Gene C with a *focused view*

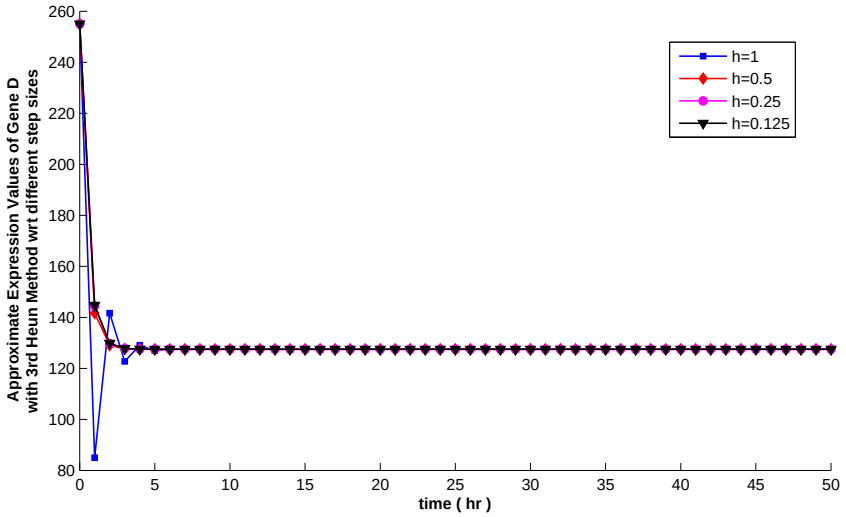


Figure 4.22: Results of Gene D using different step-sizes with 3^{rd}-order Heun's method

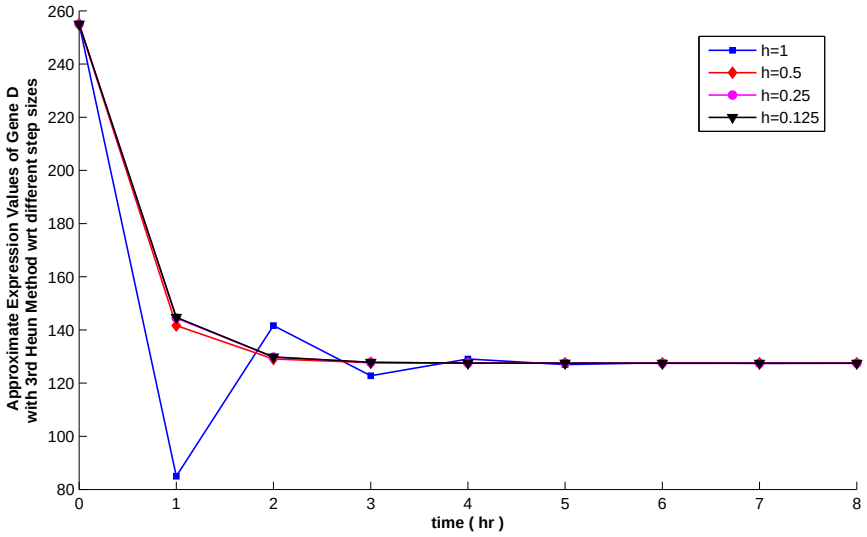


Figure 4.23: Results of Gene D with a *focused view*

(iv) Different step-size analysis with 4^{th}-order classical Runge-Kutta method

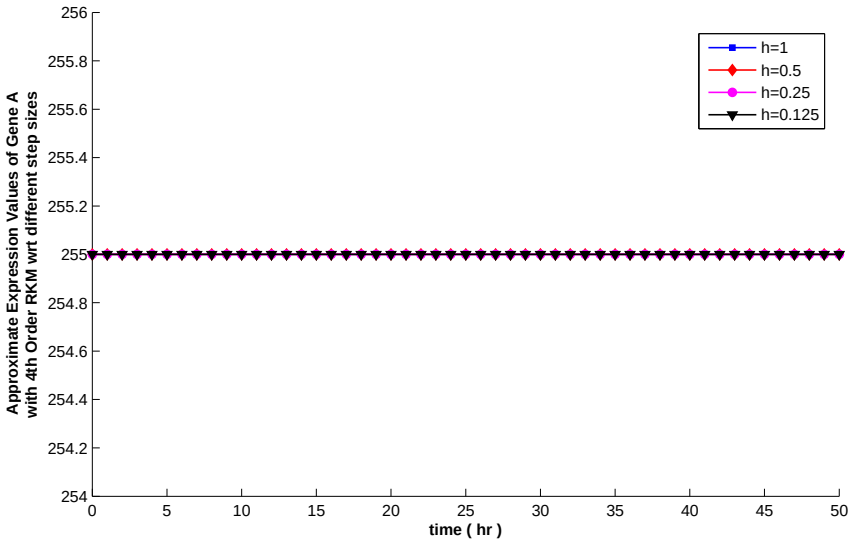


Figure 4.24: Results of Gene A using different step-sizes with 4^{th}-order classical Runge-Kutta method

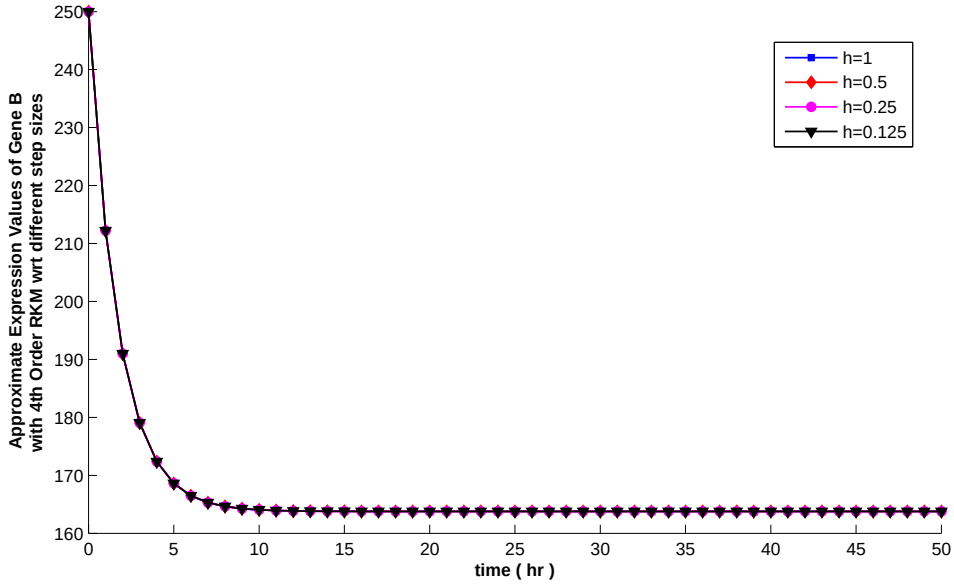


Figure 4.25: Results of Gene B using different step-sizes with 4^{th}-order classical Runge-Kutta method

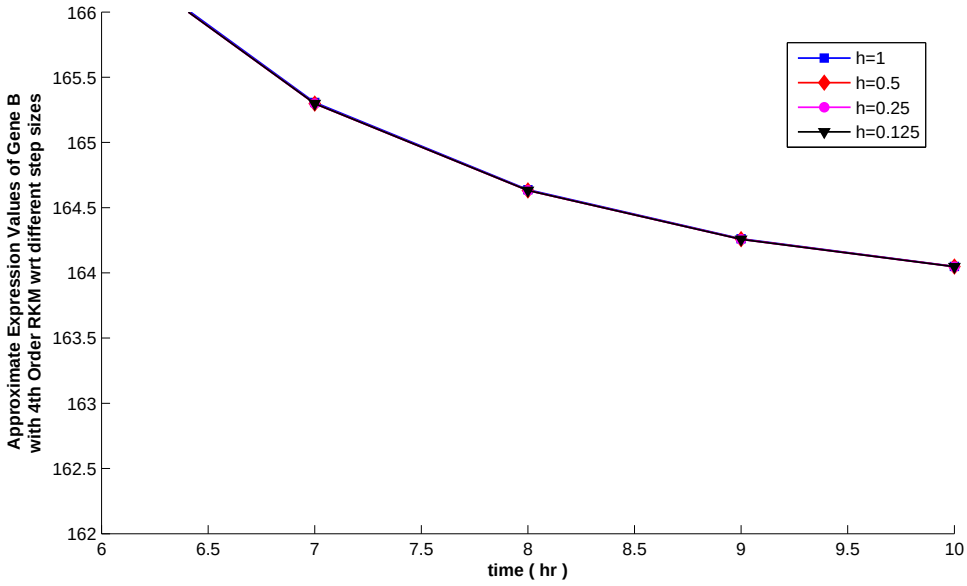


Figure 4.26: Results of Gene B with a *focused view*

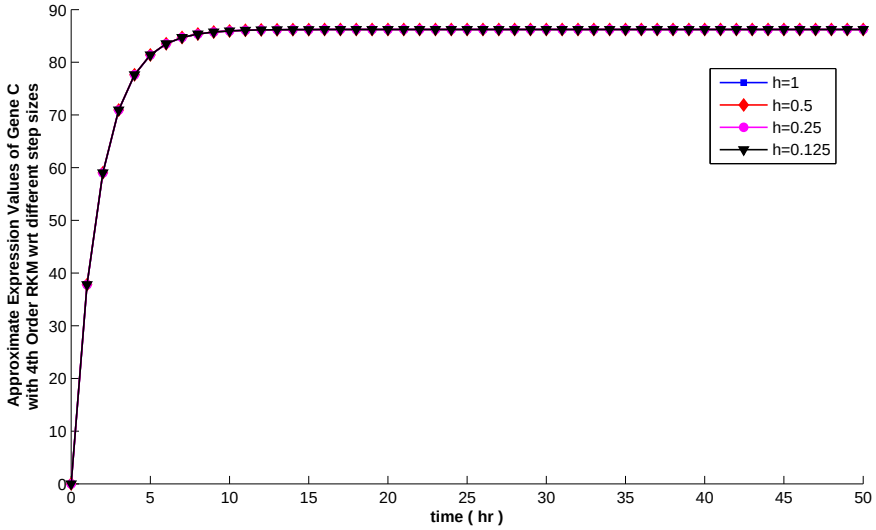


Figure 4.27: Results of Gene C using different step-sizes with 4^{th}-order classical Runge-Kutta method

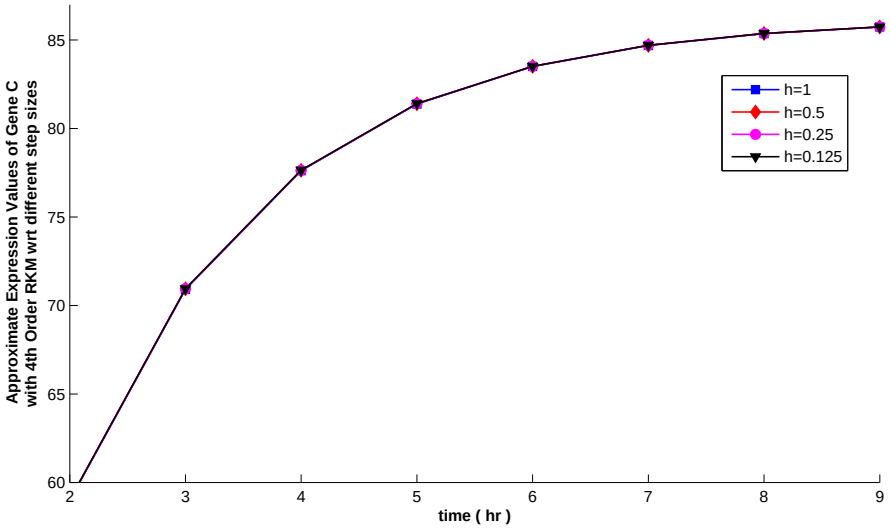


Figure 4.28: Results of Gene C with a *focused view*

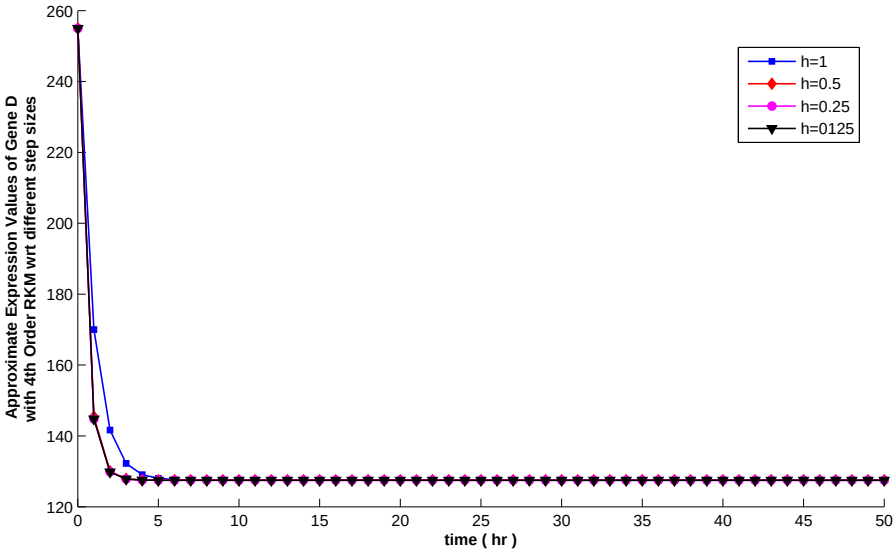


Figure 4.29: Results of Gene D using different step-sizes with 4^{th}-order classical Runge-Kutta method

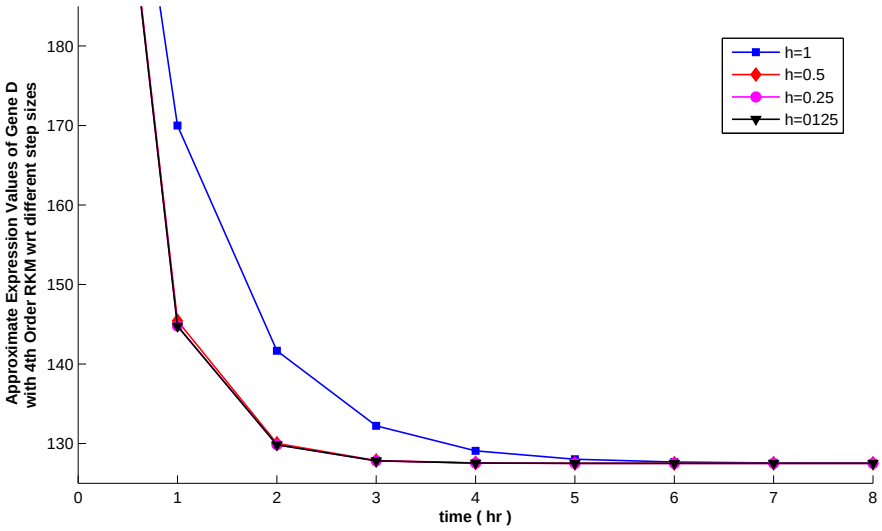


Figure 4.30: Results of Gene D with a *focused view*

According to the obtained results from *different step-size analysis* of all four numerical schemes, we can observe the following:

- When we compare obtained approximate results with the used *experimental data* that we have only at first four time levels ($t_1 = 0\text{hr}, \ldots, t_4 = 3\text{hr}$). It is seen that, for $h = 1$, Euler and 3^{rd}-order Heun's method produce approximate results which are more close to the experimental values at the first four time levels. As h_k starts to decrease like $h = 1/2, 1/4, 1/8$ then corresponding approximate values at the first time levels are getting more far from the experimental data. For the studied 2^{nd}-order and 4^{th}-order methods, the situation happens vice versa. As h_k starts to decrease, the produced approximate results are getting more close to the experimental at the first time levels.

- Here, let us remind the definition of rate of convergence:

  Suppose that we have a sequence $\{\mathbf{E}_k\}$ of iterations of gene-expression levels produced by a numerical scheme which converge to a limit point $\mathbf{E}^*$ as $k \to \infty$. Then, the ratio

  $$\mu = \limsup_{k\to\infty} \frac{|\ \mathbf{E}_{k+1} - \mathbf{E}^*\ |}{|\ \mathbf{E}_k - \mathbf{E}^*\ |}, \tag{4.27}$$

  is called *rate of convergence*. It is called *linear* if the denominator is of power one, and *quadratic* if that power is two. The rate of convergence is a constant value, $0 \leq \mu \leq 1$.

  According to this definition we look at the linear rate of convergence μ for $k = 0, \ldots, 50$ for the calculated approximate results of a fixed method for varying step-sizes. Then we compare those results by looking at them from a fixed time and see how the rate μ changes as h_k changes. We repeat the same procedure for all numerical methods that we considered.

  In this way, we perform the comparison of rate of convergence among all the numerical schemes as h decreases from 1 to $h = 1/2, 1/4, 1/8$, we observe that the calculated *linear rate of convergence* for Euler method and 3^{rd}-order Heun's method increases while for 2^{nd}-order Heun's method and 4^{th}-order classical Runge-Kutta method it decreases by a small amount, when we look at them from the same time levels (hours).

  For example: the linear rate of convergence of Euler method at time $t = 3\text{hr}$ for $h = 1$ is small than the rate for $h = 1/2$, and similarly rate for $h = 1/2$ is smaller than for $h = 1/4$ and so on.

- On the other hand as important consequence of different step-size analysis, comparison of obtained approximate results of each gene as h_k decreases among each method shows that the following:

  In Euler method, results with $h = 1$ give an alternating behavior for Gene D, but with $h = 1/2$ this behavior becomes smoother and turned to damped oscillatory behavior. As we continue to reduce the step-size h_k, the limit point in this damped oscillation is reached a few time steps before. Similarly, 3^{rd}-order Heun's method directly make smooth the results of Gene D from alternating to damped oscillatory for $h = 1$. As h_k continue to decrease, the amplitude of these oscillations becomes smaller and the limit point is reached in earlier time levels.

  Also, in 2^{nd}-order Heun's method, for Gene D, a constant behavior is obtained with $h = 1$ instead of original alternating behavior. This results can be thought as smooth but it does not goes to the fixed point and also does not show the real behavior, therefore we might obtain a ghost solution or spurious solution as described in [147]. But, when we decrease step-size as $h = 1/2, 1/4, 1/8$, then Gene D smoothly converging to the desired limit point at the end. In 4^{th}-order classical Runge-Kutta method, for gene D, smooth behavior is already obtained because of the high degree and accuracy of the method. As h_k decreases, the results are still converging to the limit point.

4.3.1.4 Testing Sensitivity

In the following, we investigate the reply of approximate results obtained from considered numerical schemes derived and listed in (4.26) (for our model $\dot{\mathbf{E}} = \mathbf{ME}$) to a various values of perturbations on the initial gene-expression data in Table 4.1. Here, we consider the choices of perturbation value ϵ as $\epsilon = 10^1, 10^0, 10^{-1}$ and 10^{-2} for these genes. We solve the same MINLP model described in (4.19), called *(OP-I)*, for the perturbed initial artificial data given below for each case of ϵ and for the same corresponding approximated derivative data in (4.24) (because of difference quotient). Then we apply all considered numerical schemes for $h_k = 1$ in order to generate approximate time series gene-expression results.

For $\epsilon = 10^1$:

$$\begin{aligned} \bar{\mathbf{E}}_1^T &= [265, \quad 260, \quad 10, \quad 265], \\ \bar{\mathbf{E}}_2^T &= [265, \quad 210, \quad 60, \quad 10], \\ \bar{\mathbf{E}}_3^T &= [265, \quad 190, \quad 80, \quad 265], \\ \bar{\mathbf{E}}_4^T &= [265, \quad 180, \quad 90, \quad 10], \end{aligned} \tag{4.28}$$

For $\epsilon = 10^0$:

$$\begin{aligned} \bar{\mathbf{E}}_1^T &= [256, \quad 251, \quad 1, \quad 256], \\ \bar{\mathbf{E}}_2^T &= [256, \quad 201, \quad 51, \quad 1], \\ \bar{\mathbf{E}}_3^T &= [256, \quad 181, \quad 71, \quad 256], \\ \bar{\mathbf{E}}_4^T &= [256, \quad 171, \quad 81, \quad 1], \end{aligned} \tag{4.29}$$

For $\epsilon = 10^{-1}$:

$$\begin{aligned} \bar{\mathbf{E}}_1^T &= [255.1, \quad 250.1, \quad 0.1, \quad 255.1], \\ \bar{\mathbf{E}}_2^T &= [255.1, \quad 200.1, \quad 50.1, \quad 0.1], \\ \bar{\mathbf{E}}_3^T &= [255.1, \quad 180.1, \quad 70.1, \quad 255.1], \\ \bar{\mathbf{E}}_4^T &= [255.1, \quad 170.1, \quad 80.1, \quad 0.1], \end{aligned} \tag{4.30}$$

For $\epsilon = 10^{-2}$:

$$\begin{aligned} \bar{\mathbf{E}}_1^T &= [255.01, \quad 250.01, \quad 0.01, \quad 255.01], \\ \bar{\mathbf{E}}_2^T &= [255.01, \quad 200.01, \quad 50.01, \quad 0.01], \\ \bar{\mathbf{E}}_3^T &= [255.01, \quad 180.01, \quad 70.01, \quad 255.01], \\ \bar{\mathbf{E}}_4^T &= [255.01, \quad 170.01, \quad 80.01, \quad 0.01]. \end{aligned} \tag{4.31}$$

By using the above perturbation values, the calculated approximate time series gene-expression results of all numerical schemes are presented in the following graphs.

(i) Perturbation analysis with Euler's method for $h_k = 1$

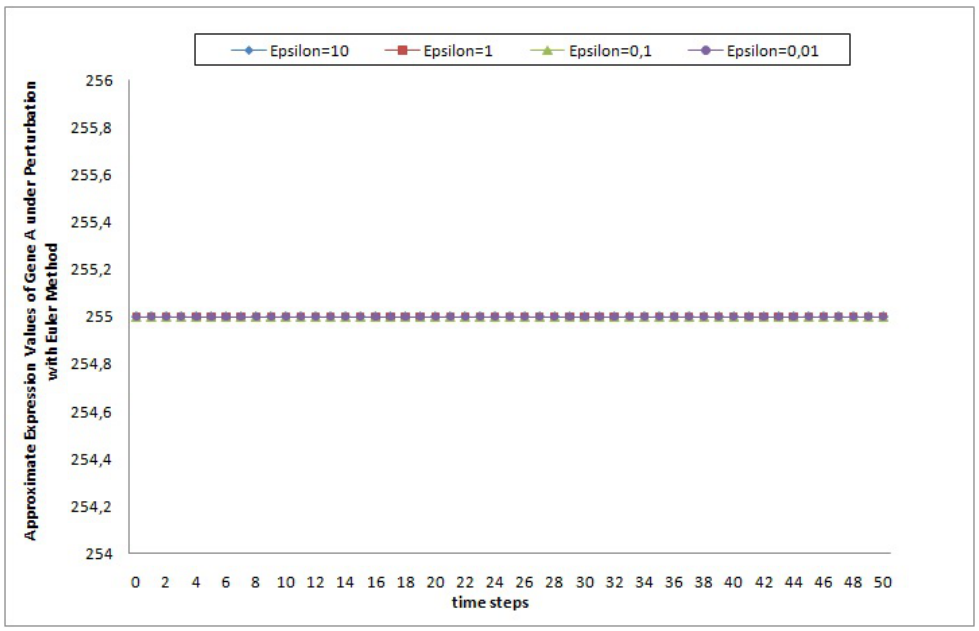


Figure 4.31: Results of Gene A with Euler method under various perturbations

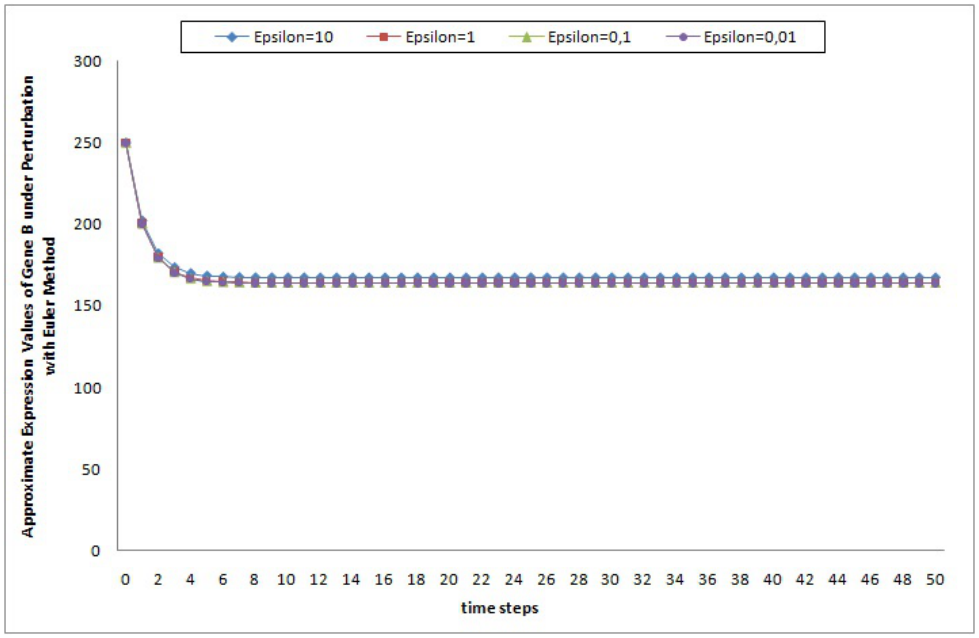


Figure 4.32: Results of Gene B with Euler method under various perturbations

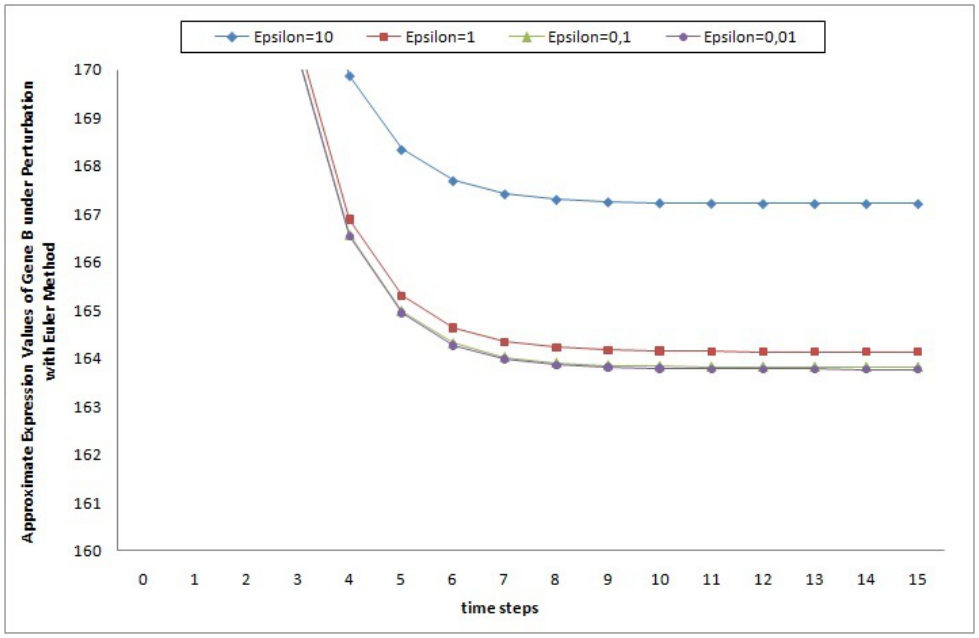


Figure 4.33: Results of Gene B with a *focused view*

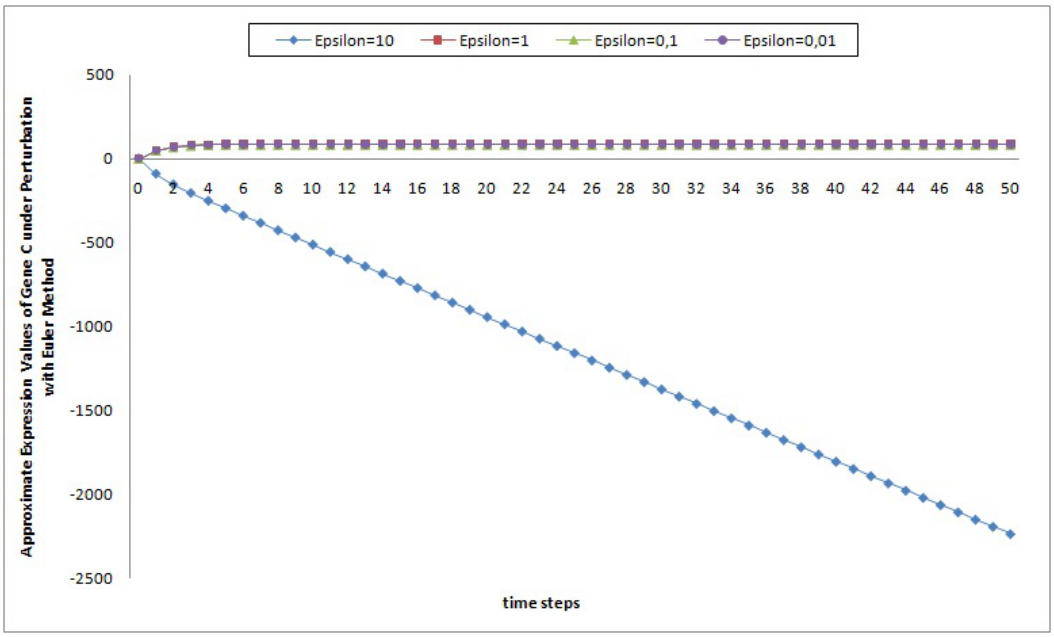


Figure 4.34: Results of Gene C with Euler method under various perturbations

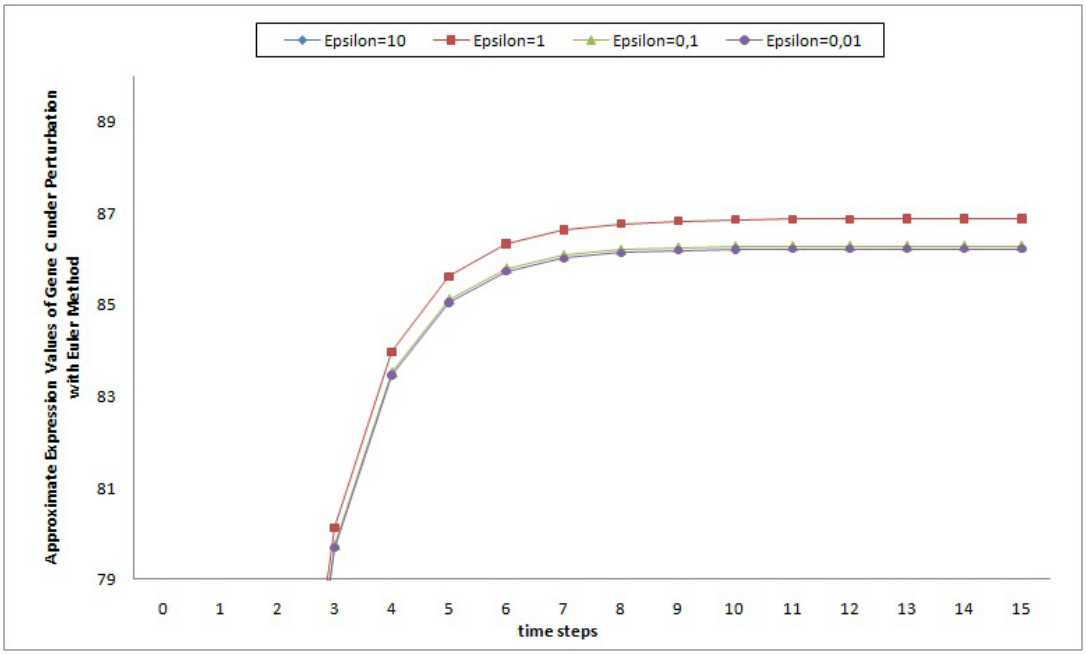


Figure 4.35: Results of Gene C with a *focused view*

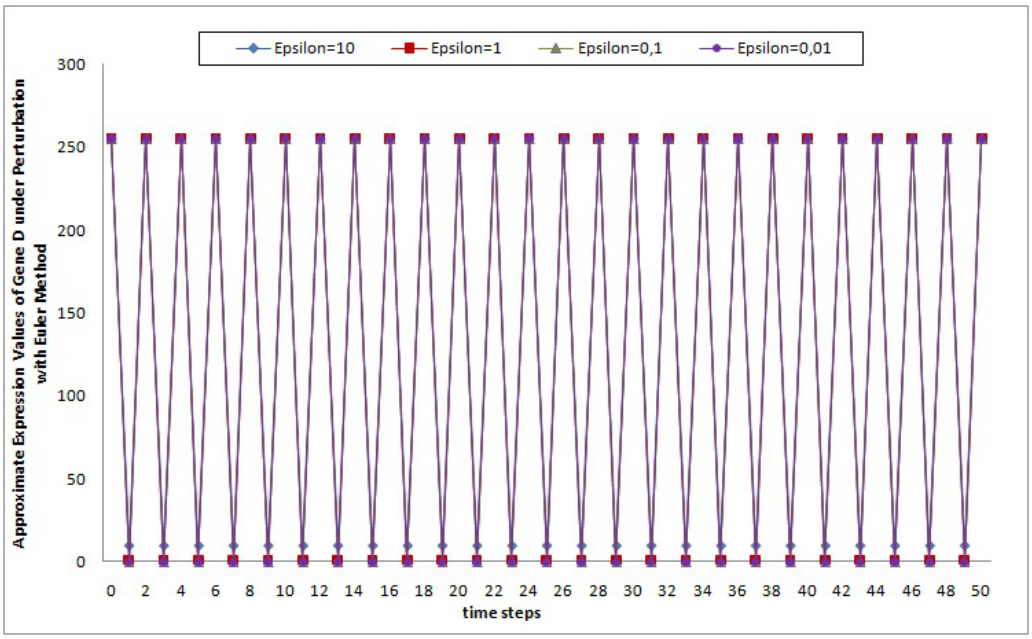


Figure 4.36: Results of Gene D with Euler method under various perturbations

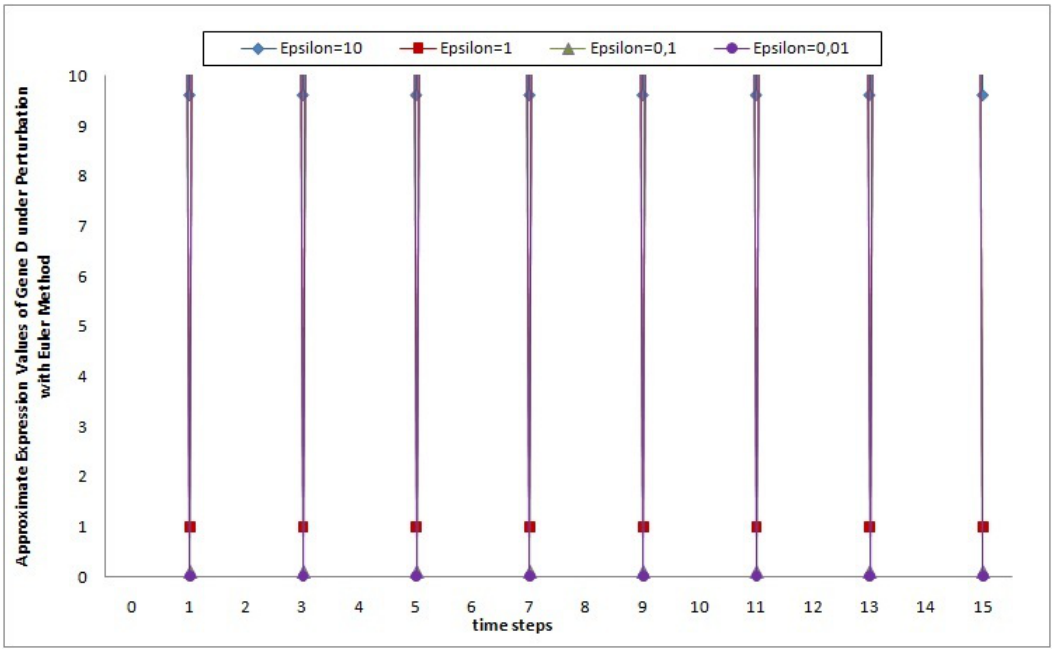


Figure 4.37: Results of Gene D with a *focused view*

(ii) Perturbation analysis with 2^{nd}-order Heun's method for $h_k = 1$

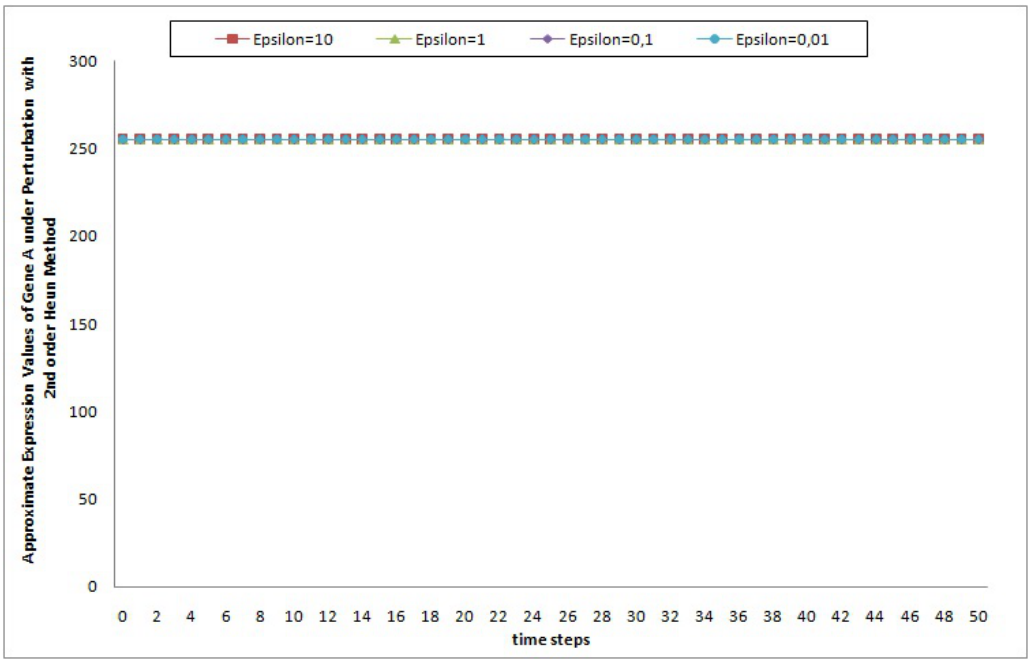


Figure 4.38: Results of Gene A with 2^{nd}-order Heun's method under various perturbations

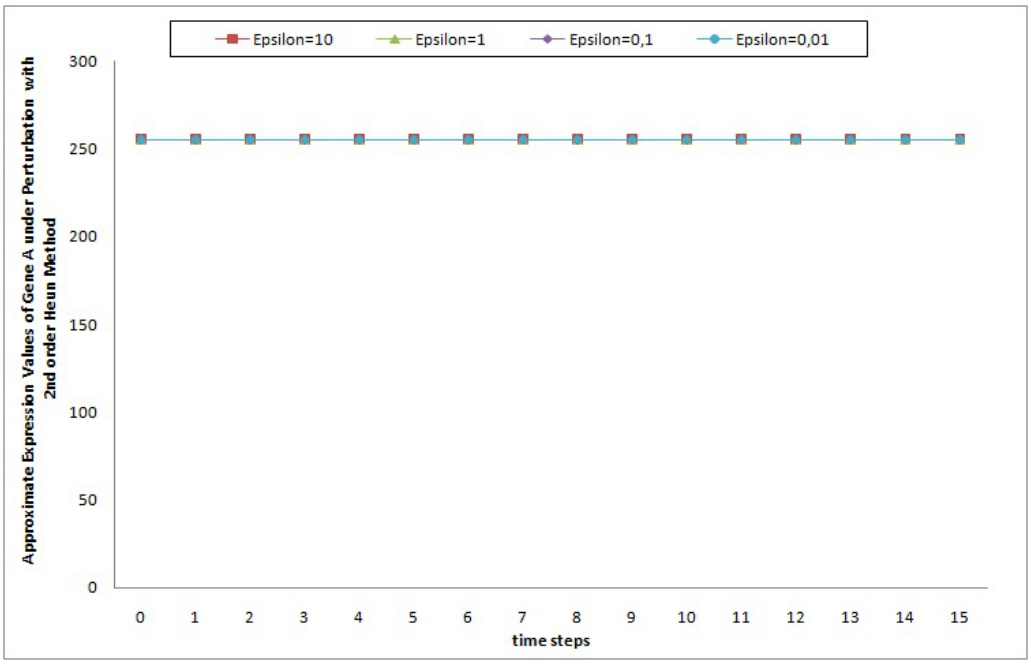


Figure 4.39: Results of Gene A with a *focused view*

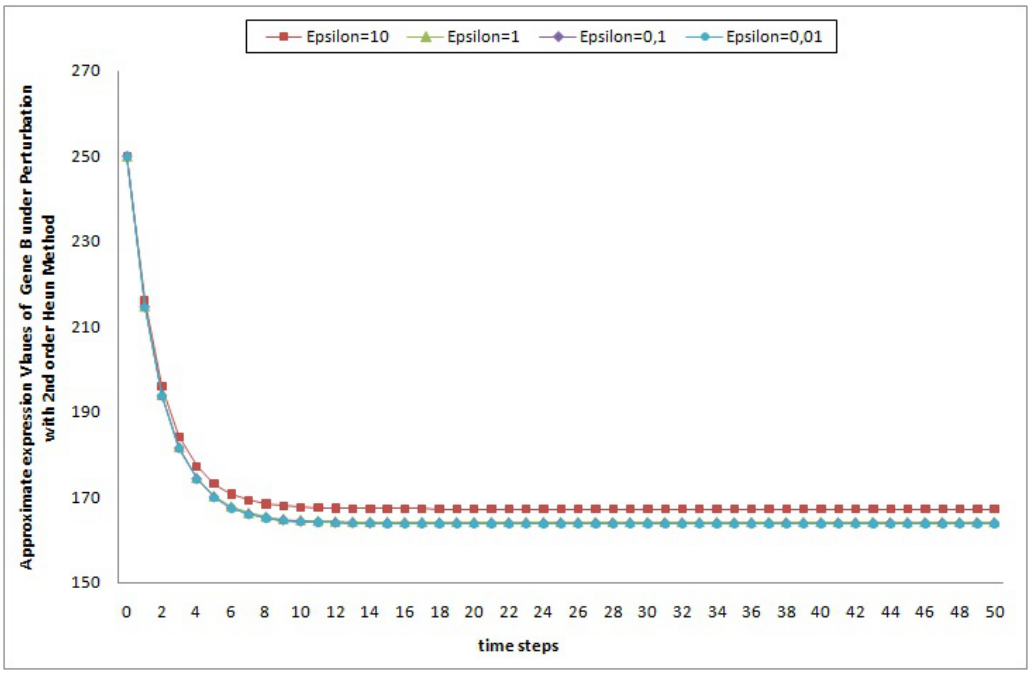


Figure 4.40: Results of Gene B with 2^{nd}-order Heun's method under various perturbations

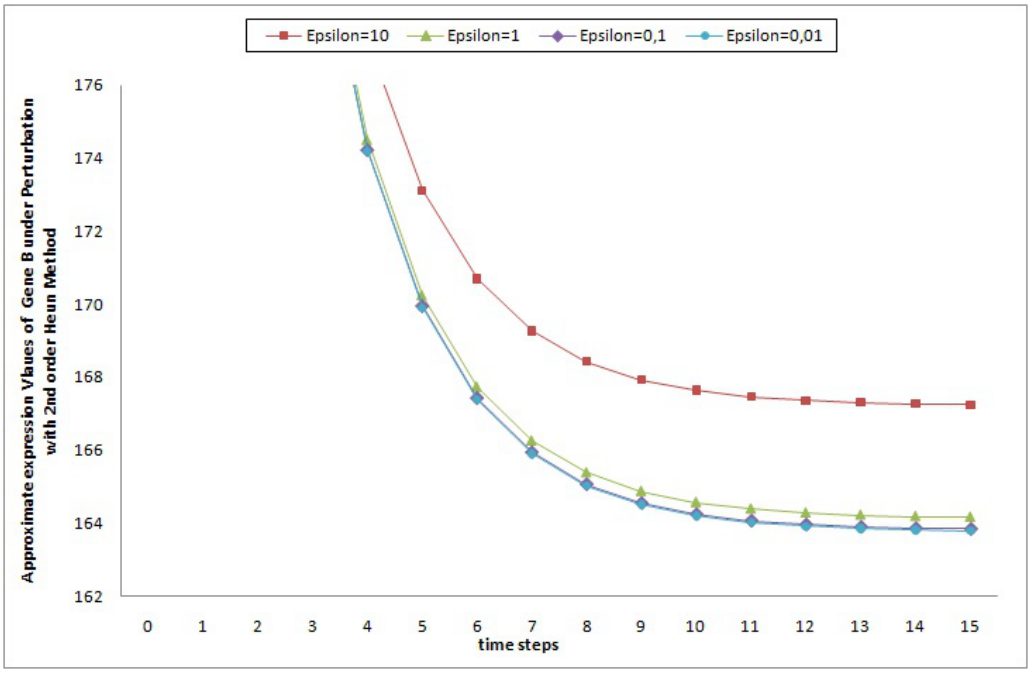


Figure 4.41: Results of Gene B with a *focused view*

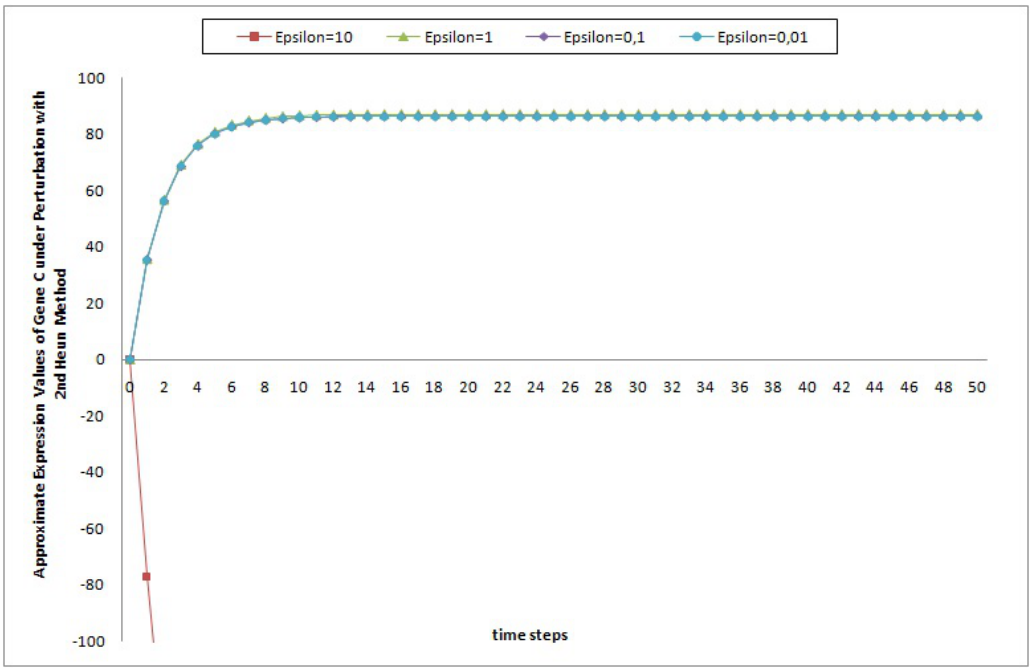


Figure 4.42: Results of Gene C with 2^{nd}-order Heun's method under various perturbations

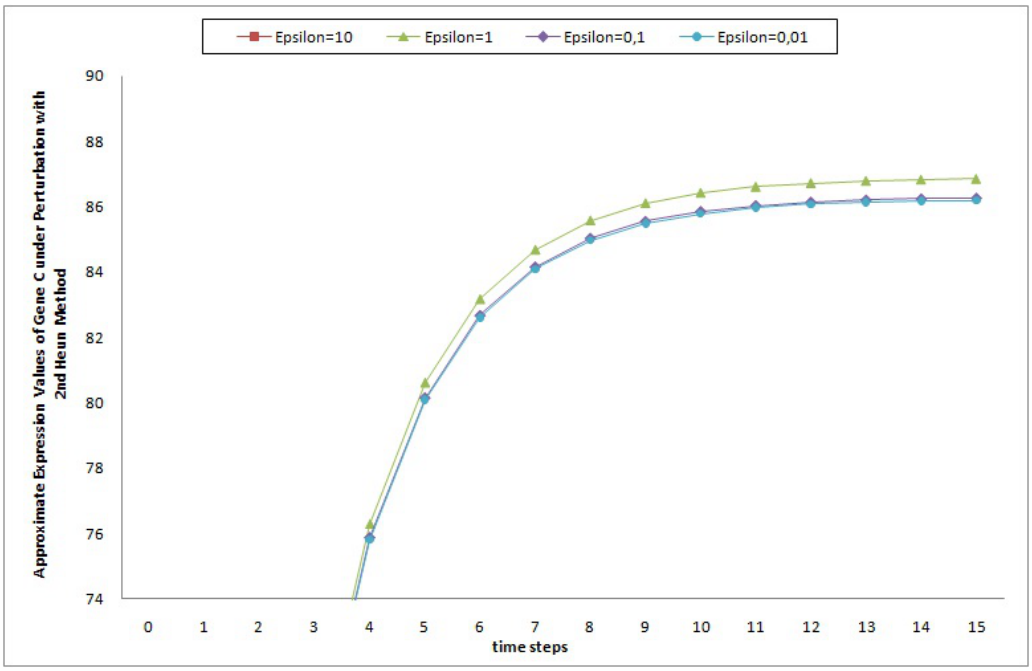


Figure 4.43: Results of Gene C with a *focused view*

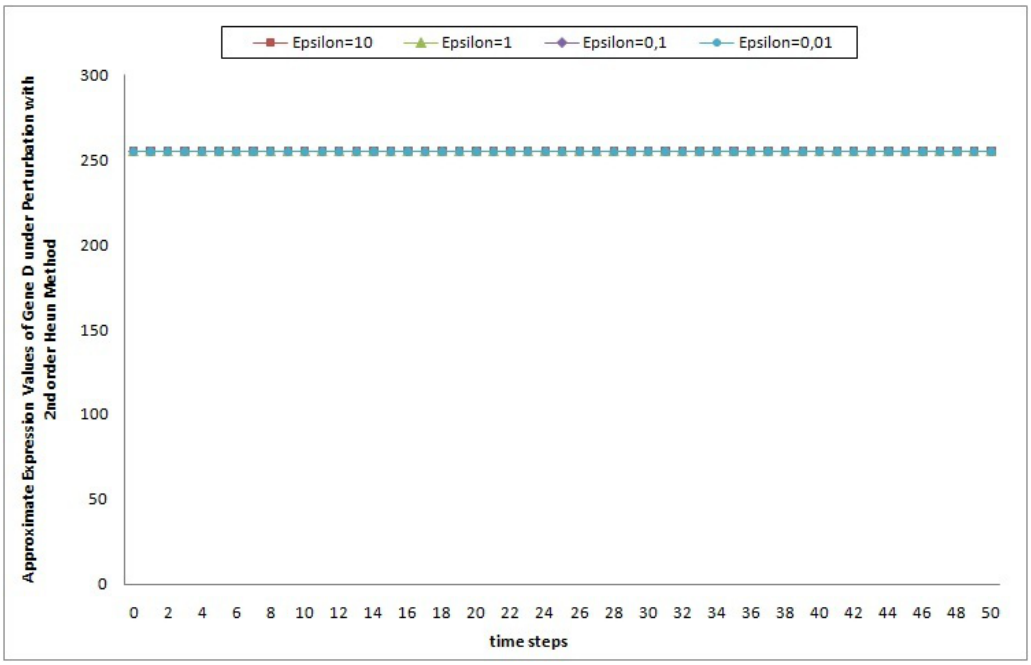


Figure 4.44: Results of Gene D with 2^{nd}-order Heun's method under various perturbations

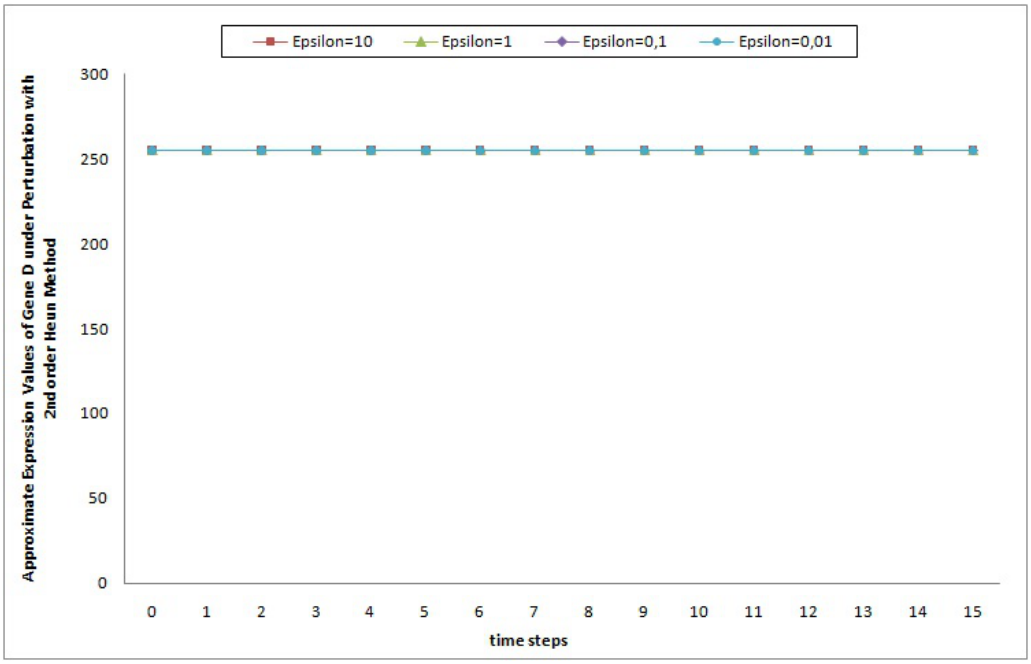


Figure 4.45: Results of Gene D with a *focused view*

(iii) Perturbation analysis with 3^{rd}-order Heun's method for $h_k = 1$

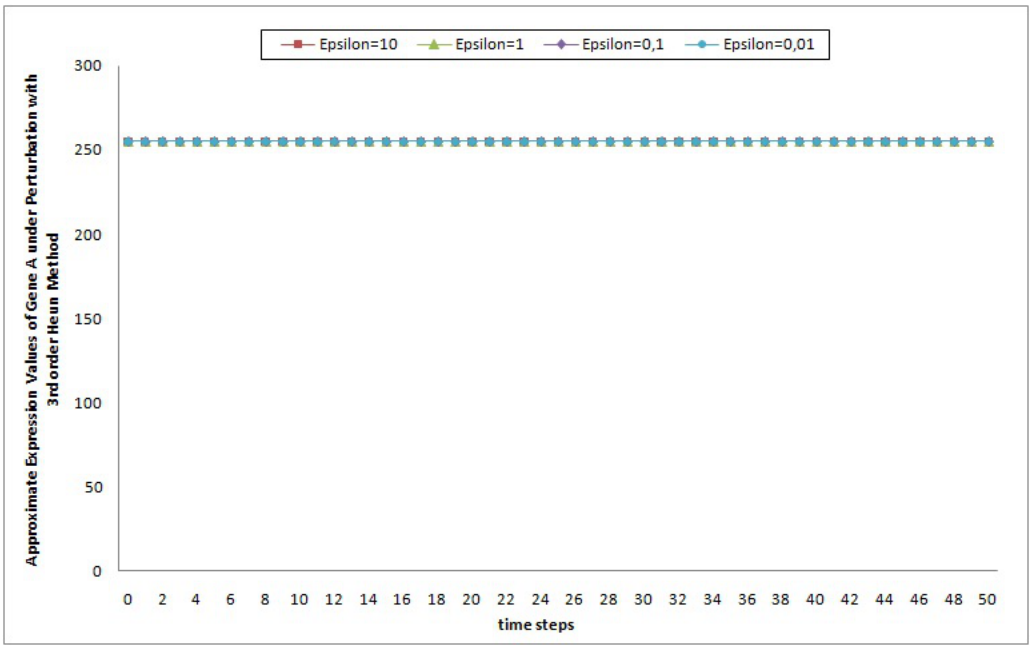


Figure 4.46: Results of Gene A with 3^{rd}-order Heun's method under various perturbations

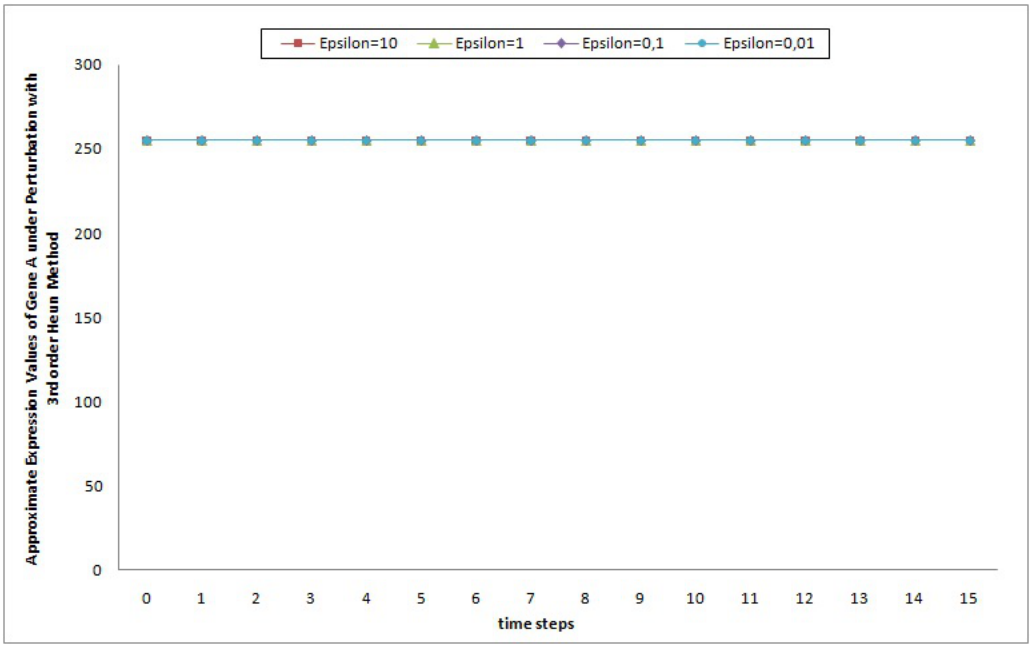


Figure 4.47: Results of Gene A with a *focused view*

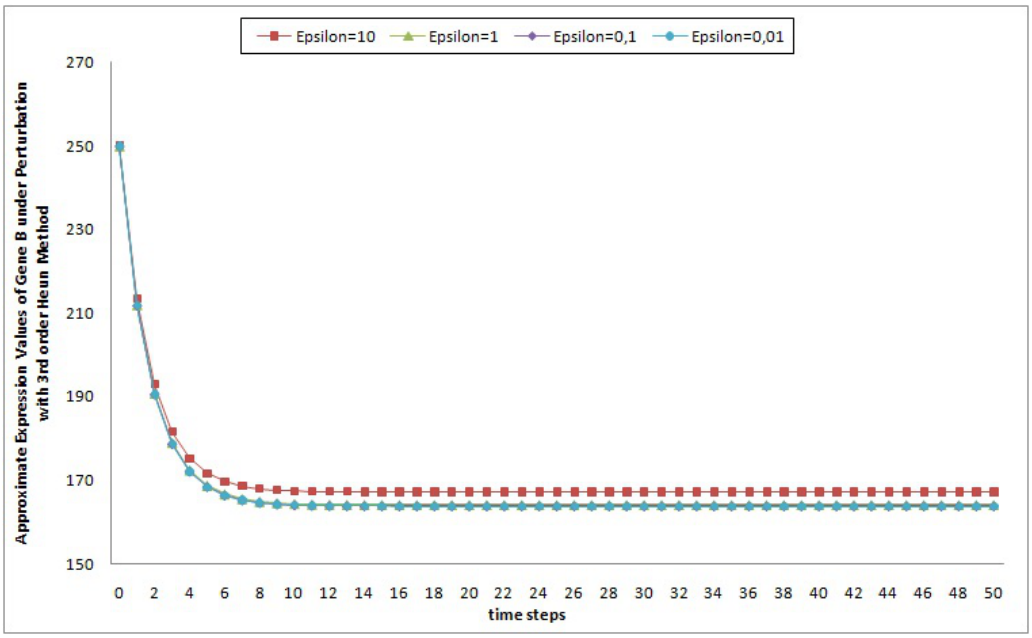


Figure 4.48: Results of Gene B with 3^{rd}-order Heun's method under various perturbations

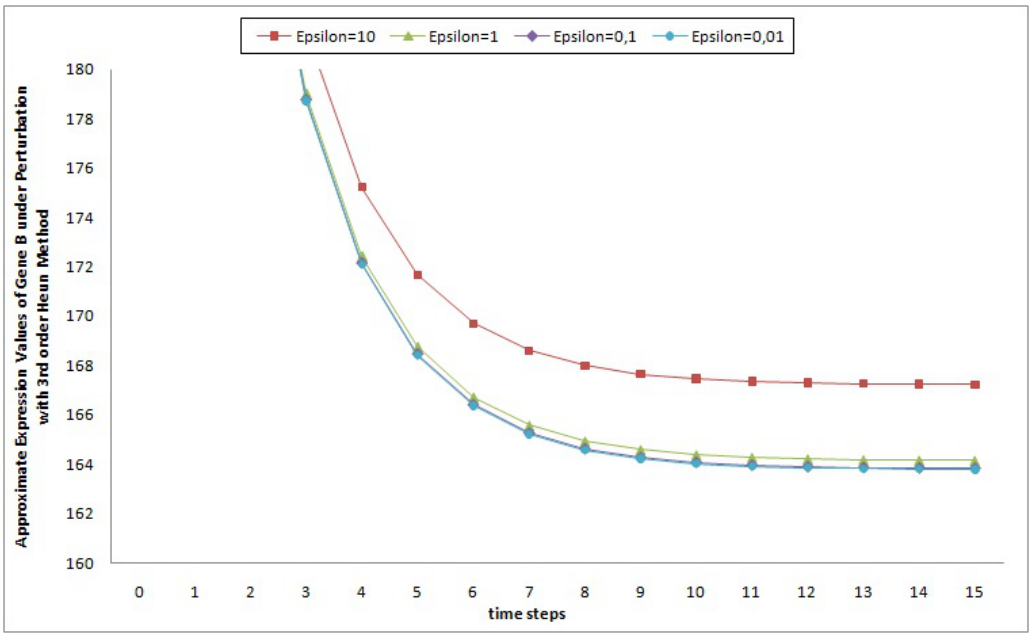


Figure 4.49: Results of Gene B with a *focused view*

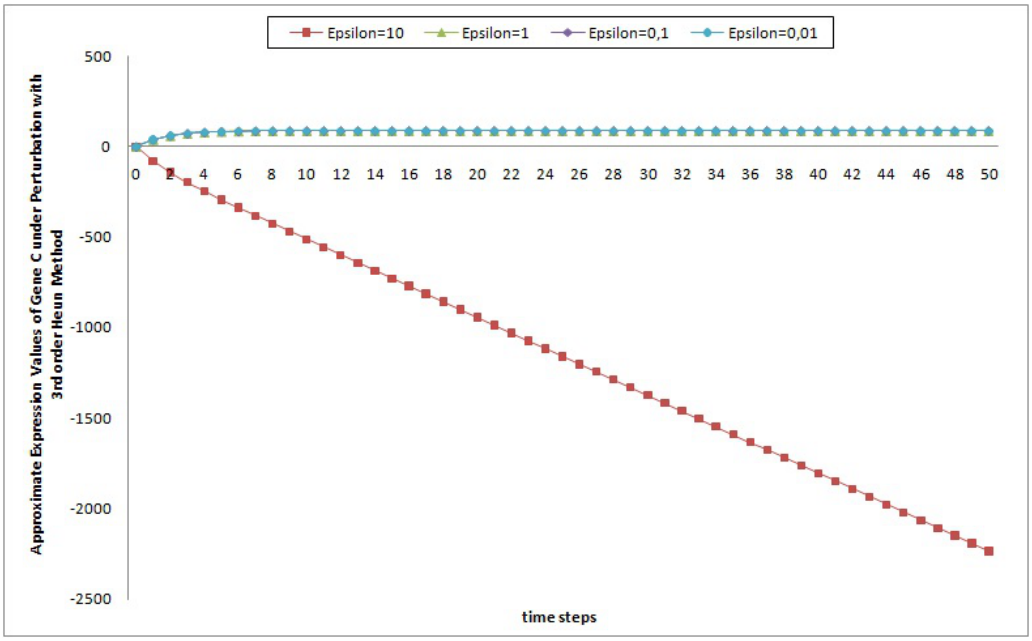


Figure 4.50: Results of Gene C with 3^{rd}-order Heun's method under various perturbations

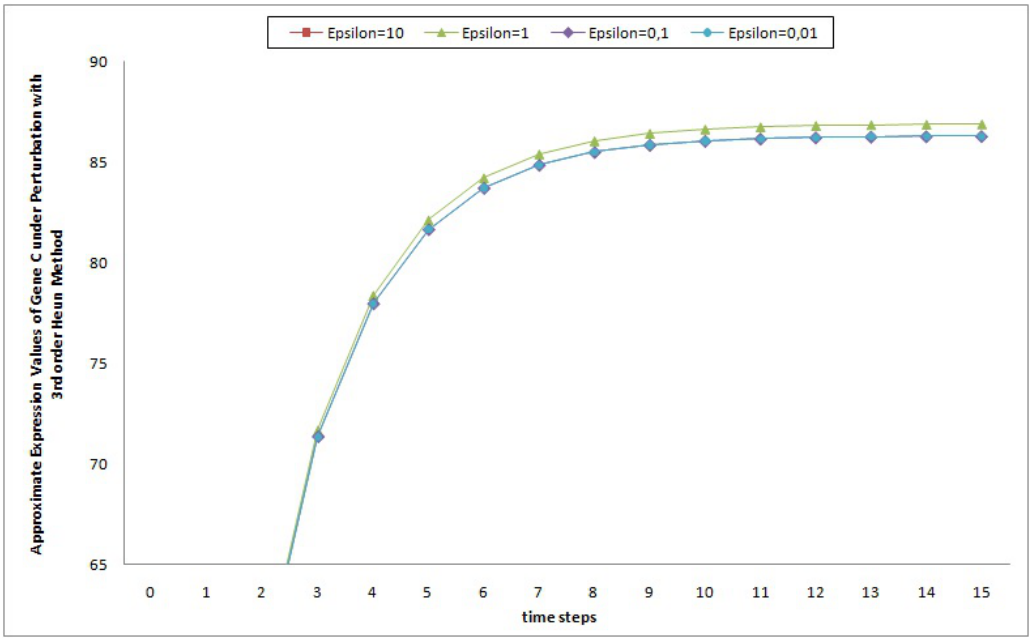


Figure 4.51: Results of Gene C with a *focused view*

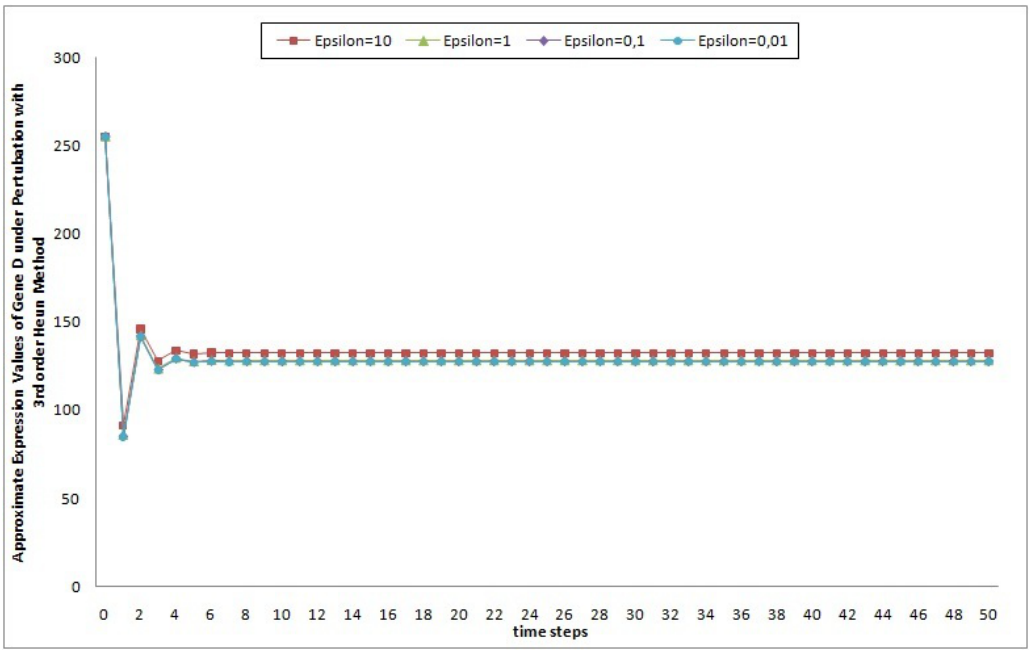


Figure 4.52: Results of Gene D with 3^{rd}-order Heun's method under various perturbations

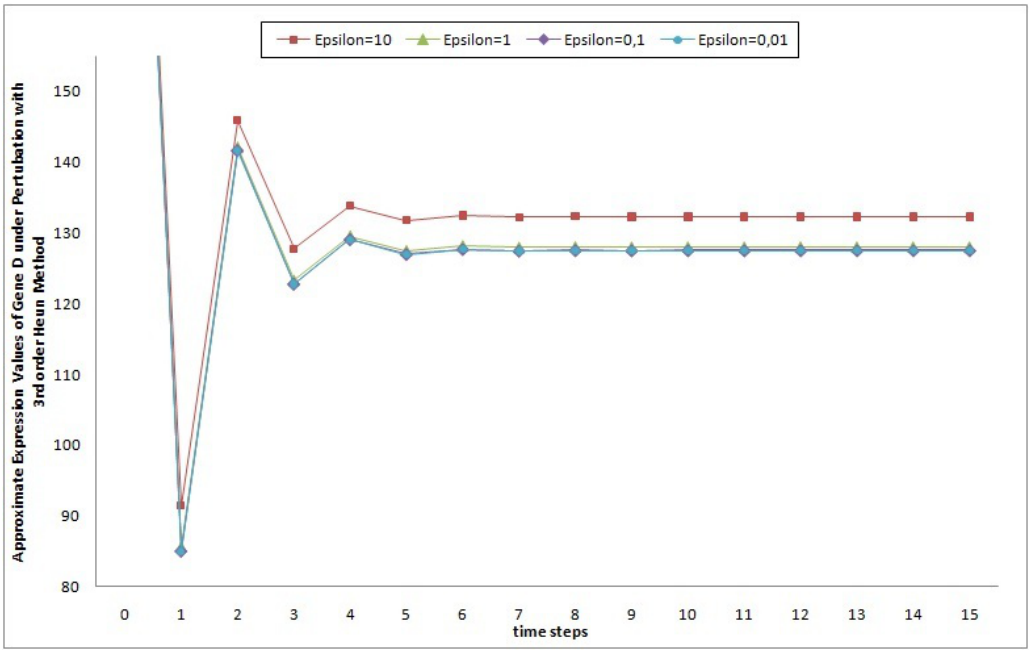


Figure 4.53: Results of Gene D with a *focused view*

(iv) Perturbation analysis with 4^{th}-order classical Runge-Kutta method for $h_k = 1$

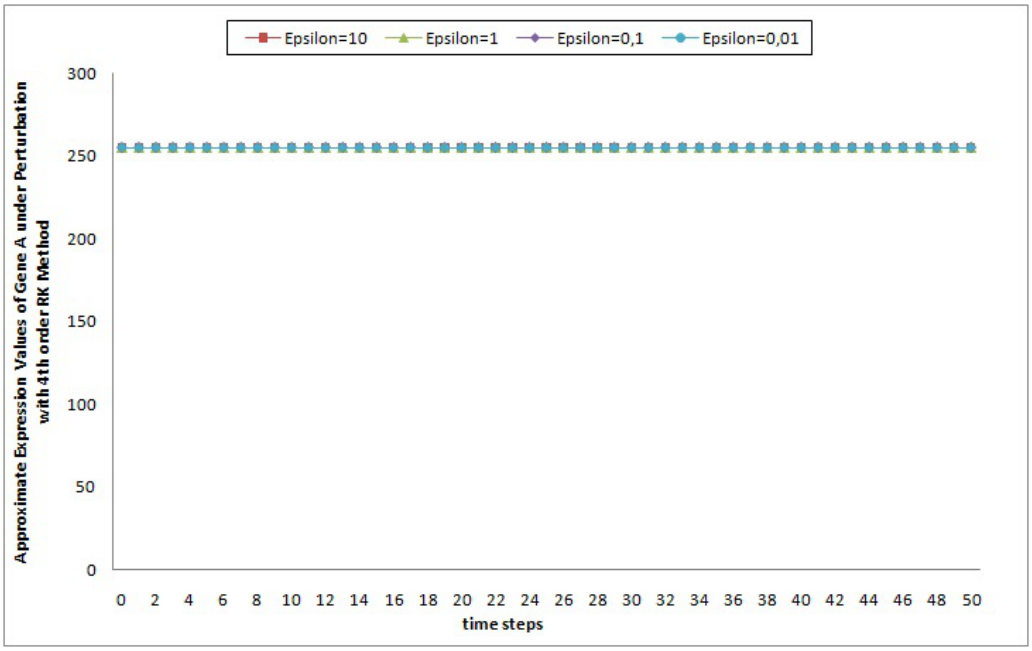


Figure 4.54: Results of Gene A with 4^{th}-order classical Runge-Kutta method under various perturbations

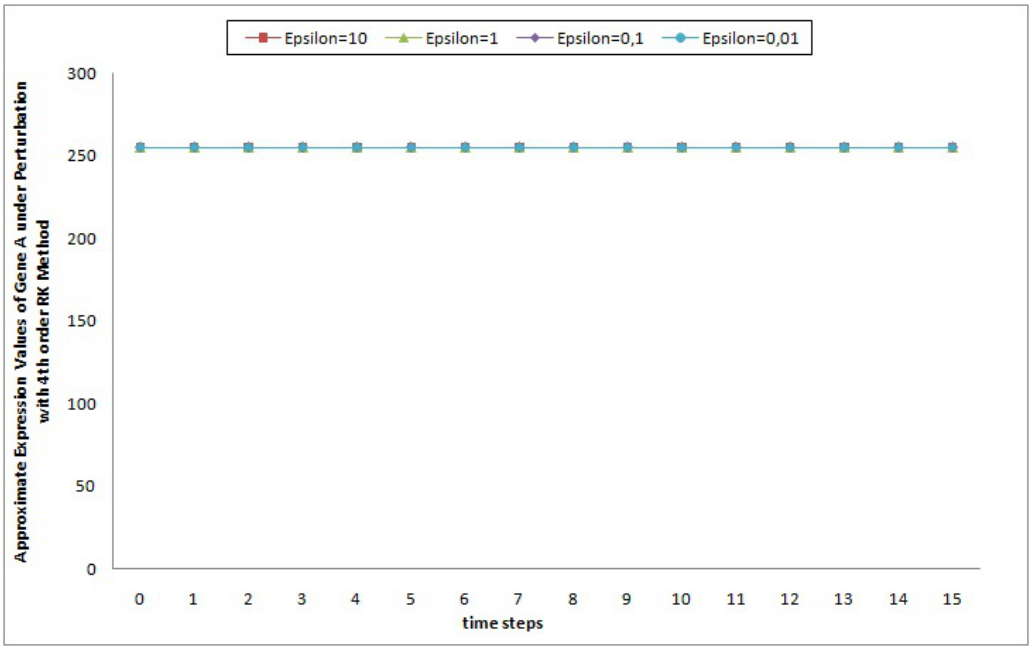


Figure 4.55: Results of Gene A with a *focused view*

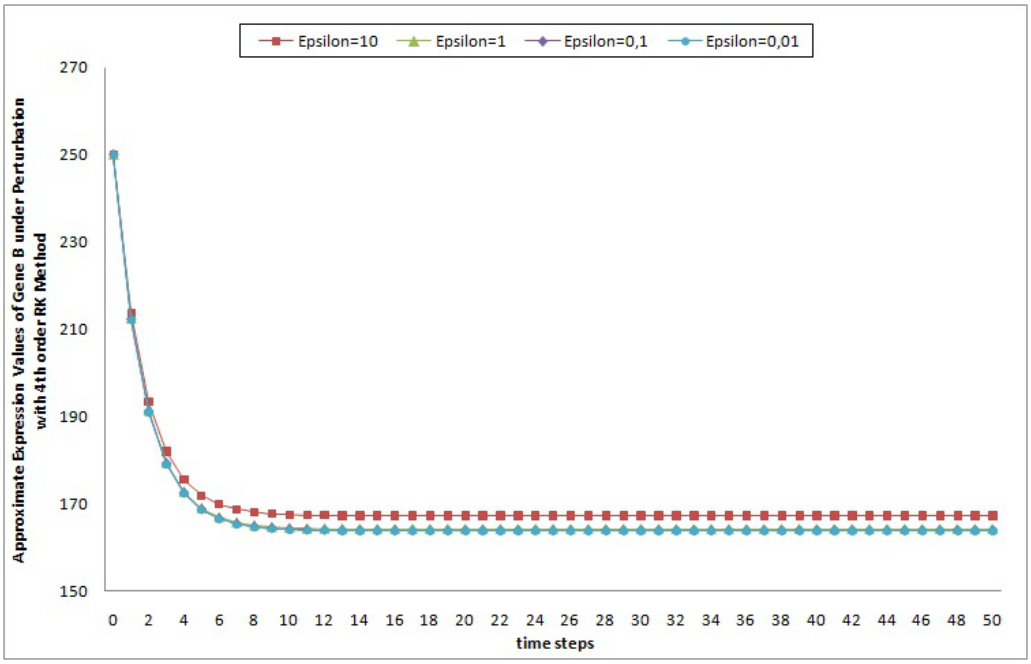


Figure 4.56: Results of Gene B with 4^{th}-order classical Runge-Kutta method under various perturbations

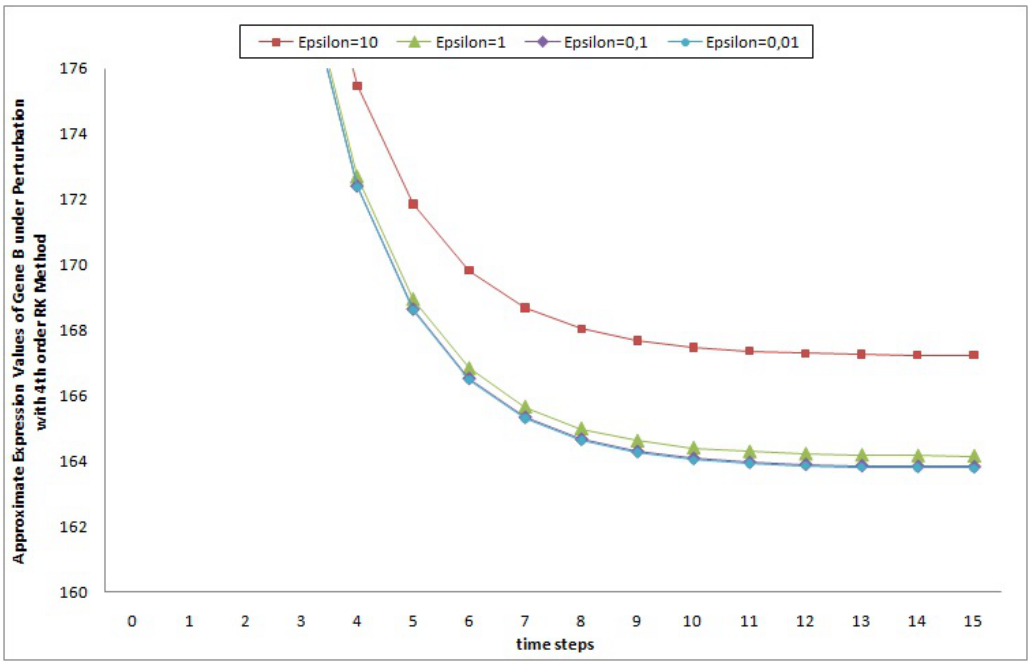


Figure 4.57: Results of Gene B with a *focused view*

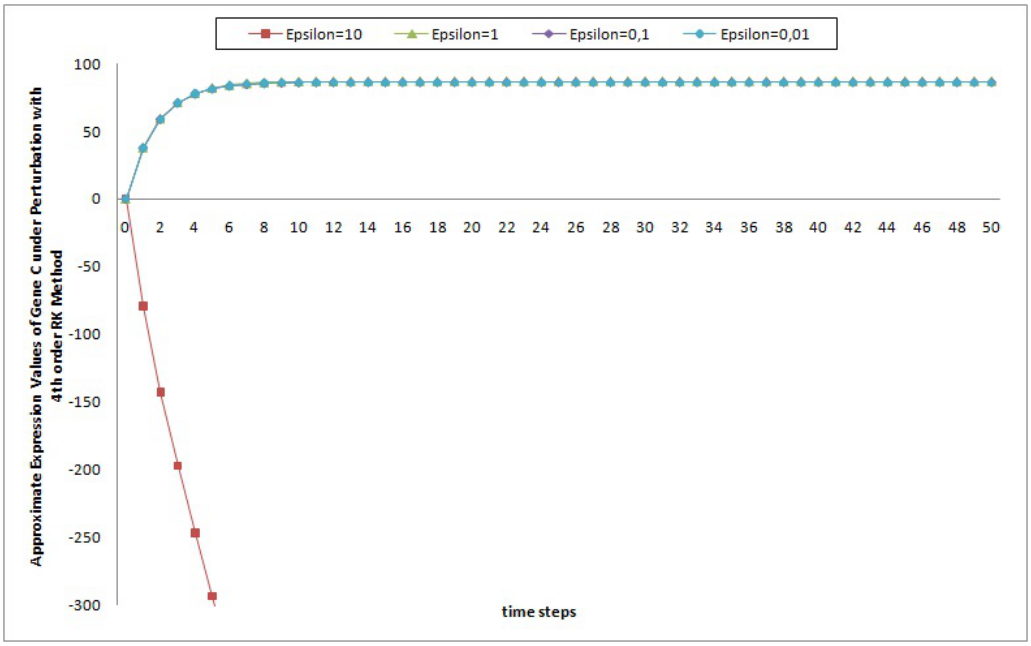


Figure 4.58: Results of Gene C with 4^{th}-order classical Runge-Kutta method under various perturbations

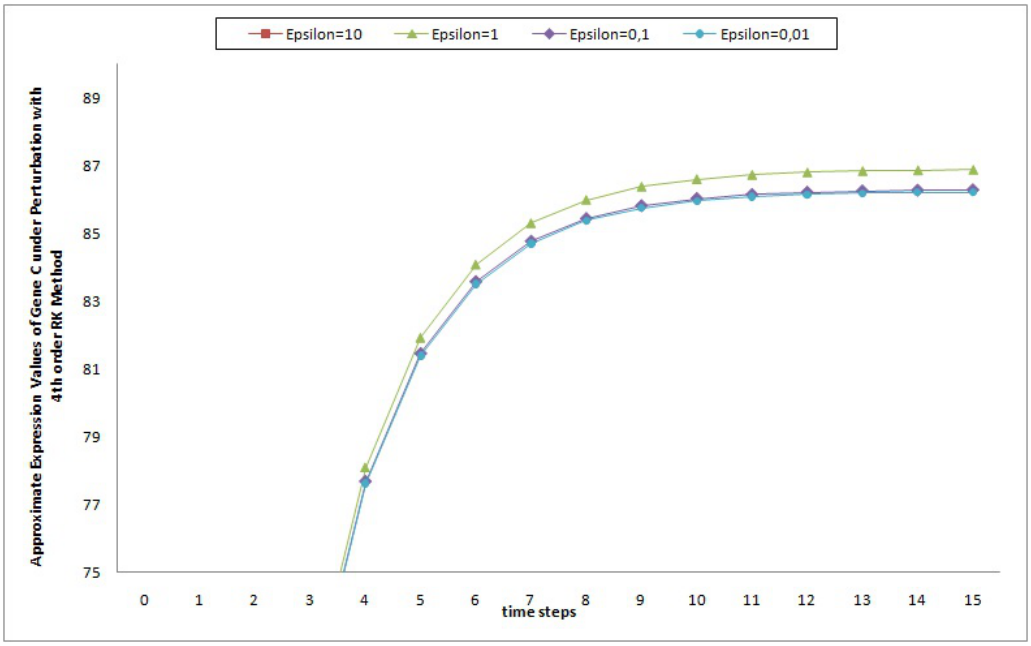


Figure 4.59: Results of Gene C with a *focused view*

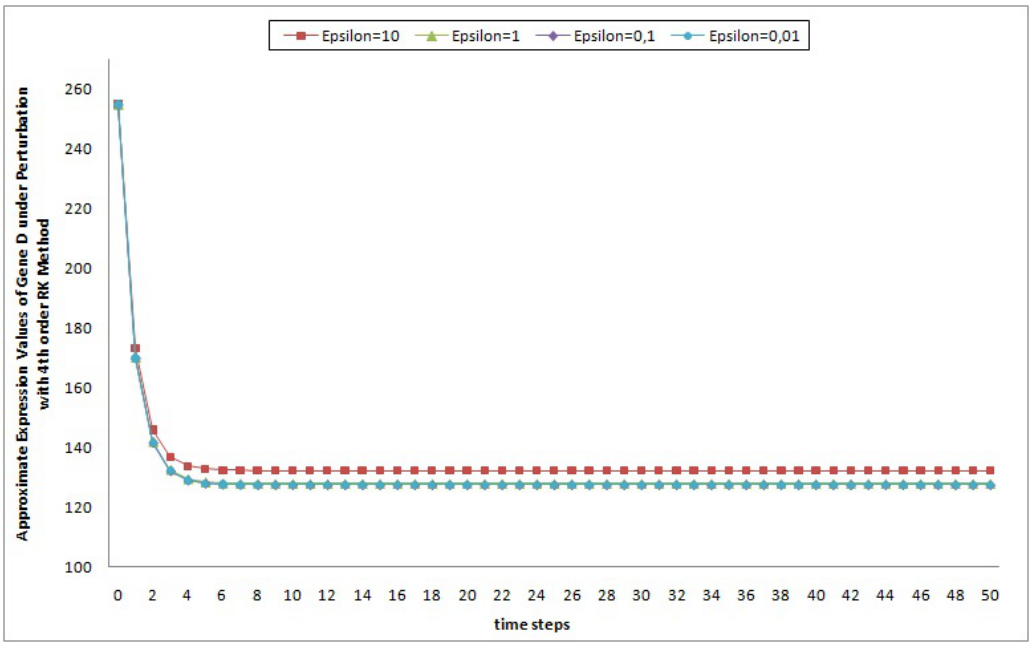


Figure 4.60: Results of Gene D with 4^{th}-order classical Runge-Kutta method under various perturbations

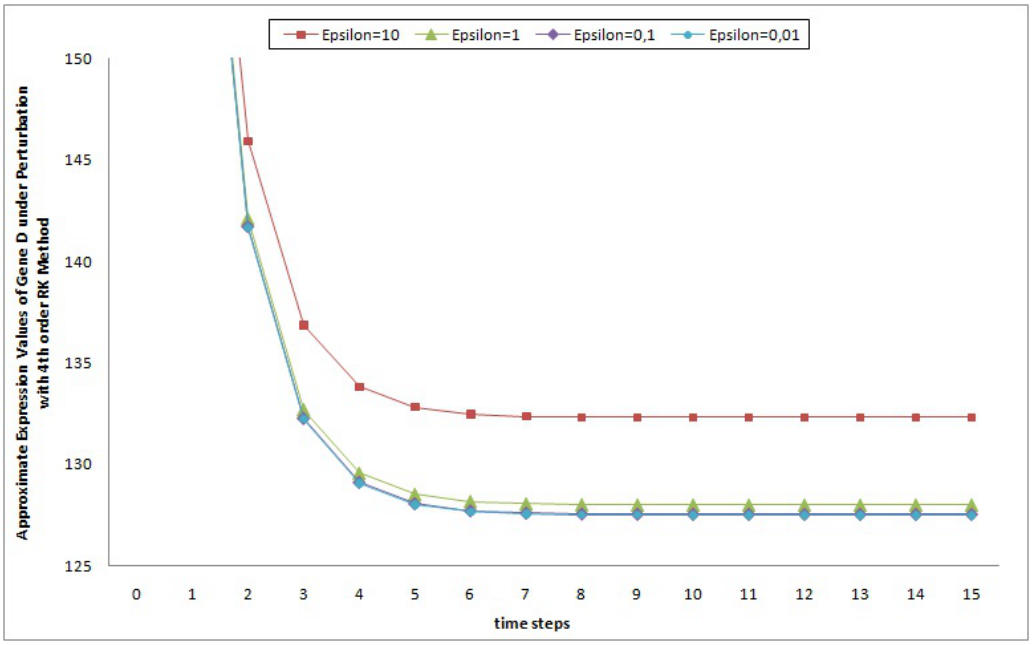


Figure 4.61: Results of Gene D with a *focused view*

According to the obtained approximate results from *perturbation analysis* of all four numerical schemes with respect to different choices of perturbation value ϵ, e.g., $10^1, 10^0, 10^{-1}, 10^{-2}$, we can observe the following:

- When we study the case $\epsilon = 10^1$, the results of all studied methods are changed somehow like a shift. But, there happens a dramatic change in Gene C which normally shows an increasing behavior. After perturbation, this gene started to take negative values in the first time levels and continue to have fast decreasing behavior.
- For the other genes in the network (Gene A, Gene B and Gene D) with $\epsilon = 10^1$ perturbation, there is a change in the results but not dramatically and in a way of preserving their existing behavior inside each considered numerical method.
- For perturbation value $\epsilon = 10^{-1}$ and $\epsilon = 10^{-2}$, obtained approximate gene-expression results are very much close to each other. Also, the results for $\epsilon = 10^0$ does not differs much. With all these three perturbation values, the following limit points are reached in each numerical scheme which is close to the limit point without perturbation given in (4.27).

$$\begin{aligned}
\mathbf{E}^* &= [255,\ 164.135967,\ 86.88391,\ 127.998047]^T, \quad \text{for} \quad \epsilon = 10^0,\\
\mathbf{E}^* &= [255,\ 163.815177,\ 86.288322,\ 127.54998]^T, \quad \text{for} \quad \epsilon = 10^{-1},\\
\mathbf{E}^* &= [255,\ 163.752423,\ 86.213687,\ 127.505]^T, \quad \text{for} \quad \epsilon = 10^{-2}.
\end{aligned}$$

4.3.1.5 Discussion

For this numerical application, we did not change the considered data and the constraints in the constructed MINLP in (4.19) in order to be able to compare our results with those of [36] where only Euler method is studied and its results are given for the data in Table 4.1. The provided data is limited in terms of the number of time points and genes, but each gene in that network shows a special behavior (constant, decreasing, increasing and alternating) both mathematically and biologically. For our next application studied in Subsection 4.3.2, we use experimentally obtained gene-expression data of a huge network belonging to a real biological phenomenon.

The detailed numerical investigation of the first illustrated example in this subsection with a

small set of artificial data shows that as the order of the applied numerical method (scheme) is increased from Euler method (as a 1^{st}-order method) to 2^{nd}-order Heun's method, 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method, which are all belong to the class of explicit Runge-Kutta methods, the following results are obtained. Briefly, we can say that:

- more smooth behaviors are obtained for the genes which shows alternating or oscillating behaviors, especially for Gene D;
- convergence to the equilibrium point is reached in all methods, sometimes with the help of decreasing the step-size or sometimes without depending on the special behavior of the gene;
- the rate of convergence is improved by reducing the step-size for the numerical schemes having odd order. Linear rate of convergence values are going to zero for some of the genes and reaching one for the others;
- the perturbation analysis that we studied results that for all genes a common allowable interval for the perturbation value ϵ can be taken as $[0, 1]$ in order to keep the general behavior of the genes among each numerical method and also to keep the convergence to the equilibrium point or its close neighbourhood. Hence, considered methods are stable under the perturbation $\epsilon \in [0, 1]$.

Regarding our performed different step-size analysis and discussion of obtained results in Subsubsection 4.3.1.3, we can say that, when using 2^{nd}-order Heun's method, or any other two or four stage (means 2^{nd}-or-4^{th}-order) explicit Runge Kutta schemes in numerical applications, one has to be very careful to choose the step-size sufficiently small in order not to obtain ghost solutions or spurious solutions, as it is shown and explained in [147, 148]. In [147], the authors stated that the two stage Heun's method is somewhat more prone such errors than other Runge Kutta methods.

4.3.2 Example with a Real-World Data Set

4.3.2.1 Data Analysis

In our real-world example, we have considered the data of mRNA transcript levels during the cell cycle of budding yeast "Saccharomyces cerevisiae" [149, 150]. These cells were collected at 17 time points taken by 10 minutes (min.) intervals in a way to cover nearly two full cell cycles containing 5-phases. The whole yeast cell cycle data set contains 6220 genes [149] and shows fluctuation of their expression levels during those 17 time points. From this data set, 416 genes are identified by Cao et al. [149] based on their peak times and they are grouped into five phases of cell cycle. Out of the 416 genes, 384 genes were classified into only one phase (class) [151].

Since we can not consider all of the genes in this clustered network because of computational complexity in the corresponding optimization problem and in numerical calculations, then we choose a small subnetwork from this huge network of 384 genes in the following way:

It is stated in [149] that, there are 25 genes as *landmarks* in the time course data and they are characterized with respect to a specific cell cycle phase. Those genes are used for specifying the cell cycle phases based on morphological markers. Among these genes we could find the data of 23 (from the whole data given at 'http://faculty.washington.edu/kayee/model/') which are covering all phases of the cell. Since they are considered as landmarks, this means that they are highly expressed genes in their cell cycle phase. Beyond the selected 23 genes, we take 3 more genes as *house keeping* genes that listed in Table 4.4 with their functional properties. The reason of including these genes can be given as follows: Each living organism needs energy absorbed from the environment in order to perform the basic metabolism, like the chemical events acquiring energy. Genes encoding for proteins which are involved in such basic metabolism are called *house keeping genes*. Those genes are usually expressed at a constant level and they are therefore a good tool to normalize gene expression data [152].

We identify and select those 3 house keeping genes by using a powerful and user-friendly biological web-application software called *OrfMapper* [152] (we refer to [152, 153] for details). Therefore, a collection of 26 genes that we selected as our small subnetwork are listed with their explanations in Table 4.3 (extracted from [149]) and Table 4.4 given below:

Table 4.3: Information about selected 26 genes in the network [149]

Name of gene	Cell-cycle phase	Functional explanation
YLR079w	Early G_1 phase	Cell cycle regulation
YJL194w	Early G_1 phase	DNA replication
YLR274w	Early G_1 phase	DNA replication
YBR202w	Early G_1 phase	DNA replication
YGR109c	Late G_1 phase	Cell cycle regulation
YPR120c	Late G_1 phase	Cell cycle regulation
YPL256c	Late G_1 phase	Cell cycle regulation
YMR199w	Late G_1 phase	Cell cycle regulation
YER070w	Late G_1 phase	DNA replication
YOR074c	Late G_1 phase	DNA replication
YDL164c	Late G_1 phase	DNA replication
YNL126w	S phase	Chromosome segregation
YHR172w	S phase	Chromosome segregation
YBL003c	S phase	DNA replication
YBL002w	S phase	DNA replication
YKL049c	G_2 phase	Chromosome segregation
YCL014w	G_2 phase	Directional growth
YGR108w	M phase	Cell cycle regulation
YPR119w	M phase	Cell cycle regulation
YAL040c	M phase	Cell cycle regulation
YGR092w	M phase	Chromosome segregation
YDR146c	M phase	Transcription factors
YLR131c	M phase	Transcription factors
YCR005c	Early G_1 phase	Housekeeping genes
YCL040w	Early G_1 phase	Housekeeping genes
YNR016c	Early G_1 phase	Housekeeping genes

Table 4.4: Explanations for the selected housekeeping genes among 26 genes

Gene ID	Metabolism	Enzyme Name
YCR005C	Citryte Cycle	Non-mitochondrial citrate synthase
YCL040W	Glycolysis	Glucose phosphorylation
YNR016C	Pyruvate metabolism	Acetyl-CoA carboxylase

The raw data [149] of 26 genes in our selected subnetwork is presented in the following together with the correspondingly calculated approximate derivative data using the forward difference approximation given in Eq. (3.25), and taking 10 minutes difference between the time intervals in terms of hours, e.g., $h_k = 1/6$ hour (hr.).

Table 4.5: Experimental raw data of selected 26 genes along 17 time points per 10 minutes

Name of Gene/Time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}
YLR079w	352	295	355	308	361	356	294	212	541	1286	813	595	490	398	400	389	692
YJL194w	78	246	121	88	115	149	130	108	231	265	138	162	130	151	107	123	215
YLR274w	209	262	231	140	148	165	184	271	416	465	351	243	198	162	179	252	324
YBR202w	114	273	179	125	115	145	289	491	478	530	369	218	187	176	295	483	411
YGR109c	174	333	747	312	110	120	144	88	104	392	443	297	193	197	142	112	144
YPR120c	562	785	1756	949	659	612	467	563	455	1207	1325	643	647	657	509	464	508
YPL256c	445	1620	2485	1303	916	808	738	636	303	791	1288	1624	1035	839	687	868	433
YMR199w	149	1045	1344	1305	998	1013	717	571	323	1152	1870	1388	989	1074	769	614	601
YER070w	78	502	823	456	279	197	123	90	86	334	513	524	436	313	177	163	168
YOR074c	17	59	106	93	57	37	18	18	19	28	82	90	89	36	19	12	15
YDL164c	216	356	817	500	267	242	216	172	221	627	740	535	397	418	270	239	269
YNL126w	142	147	188	238	244	279	205	123	94	56	110	153	236	211	160	91	104
YHR172w	57	83	133	135	106	121	85	52	50	53	68	73	101	89	56	49	33
YBL003c/	723	553	1597	1753	2478	887	1362	1053	745	1205	1569	1916	1415	1720	2300	1579	1156
YBL002w/	278	470	1444	1980	2384	868	962	877	594	850	1563	2265	1845	1982	1677	1595	968
YKL049c	255	250	465	516	595	494	427	390	329	341	382	440	487	548	521	430	357
YCL014w	39	38	32	54	97	120	162	182	120	28	47	47	71	100	93	58	89
YGR108w	49	124	90	92	149	288	391	338	323	158	134	125	148	270	340	338	310
YPR119w	92	158	122	145	259	348	489	554	422	236	185	151	218	267	365	350	365
YAL040c	1085	1013	752	750	1274	707	1376	1294	1140	2800	1352	1103	844	1087	1261	1240	1041
YGR092w	113	131	134	120	167	223	321	371	376	274	223	209	167	181	192	286	286
YDR146c	121	167	272	268	323	540	634	591	606	540	463	357	449	937	743	1024	659
YLR131c	148	113	144	209	319	360	434	411	462	398	226	187	325	433	343	466	344
YCR005c	439	250	150	172	147	143	232	500	584	506	517	655	629	436	424	281	307
YCL040w	813	302	167	229	297	208	184	283	350	659	823	682	480	539	512	484	854
YNR016c	1074	1517	1956	1619	1902	1202	1171	1011	932	1704	1464	1287	962	1139	1046	1238	1166

Table 4.6: Approximated derivative raw data of selected 26 genes

Name of Gene/Time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}
YLR079w	-342	360	-282	318	-30	-372	-492	1974	4470	-2838	-1308	-630	-552	12	-66	1818
YJL194w	1008	-750	-198	162	204	-114	-132	738	204	-762	144	-192	126	-264	96	552
YLR274w	318	-186	-546	48	102	114	522	870	294	-684	-648	-270	-216	102	438	432
YBR202w	954	-564	-324	-60	180	864	1212	-78	312	-966	-906	-186	-66	714	1128	-432
YGR109c	954	2484	-2610	-1212	60	144	-336	96	1728	306	-876	-624	24	-330	-180	192
YPR120c	1338	5826	-4842	-1740	-282	-870	576	-648	4512	708	-4092	24	60	-888	-270	264
YPL256c	7050	5190	-7092	-2322	-648	-420	-612	-1998	2928	2982	2016	-3534	-1176	-912	1086	-2610
YMR199w	5376	1794	-234	-1842	90	-1776	-876	-1488	4974	4308	-2892	-2394	510	-1830	-930	-78
YER070w	2544	1926	-2202	-1062	-492	-444	-198	-24	1488	1074	66	-528	-738	-816	-84	30
YOR074c	252	282	-78	-216	-120	-114	0	6	54	324	48	-6	-318	-102	-42	18
YDL164c	840	2766	-1902	-1398	-150	-156	-264	294	2436	678	-1230	-828	126	-888	-186	180
YNL126w	30	246	300	36	210	-444	-492	-174	-228	324	258	498	-150	-306	-414	78
YHR172w	156	300	12	-174	90	-216	-198	-12	18	90	30	168	-72	-198	-42	-96
YBL003c/	-1020	6264	936	4350	-9546	2850	-1854	-1848	2760	2184	2082	-3006	1830	3480	-4326	-2538
YBL002w/	1152	5844	3216	2424	-9096	564	-510	-1698	1536	4278	4212	-2520	822	-1830	-492	-3762
YKL049c	-30	1290	306	474	-606	-402	-222	-366	72	246	348	282	366	-162	-546	-438
YCL014w	-6	-36	132	258	138	252	120	-372	-552	114	0	144	174	-42	-210	186
YGR108w	450	-204	12	342	834	618	-318	-90	-990	-144	-54	138	732	420	-12	-168
YPR119w	396	-216	138	684	534	846	390	-792	-1116	-306	-204	402	294	588	-90	90
YAL040c	-432	-1566	-12	3144	-3402	4014	-492	-924	9960	-8688	-1494	-1554	1458	1044	-126	-1194
YGR092w	108	18	-84	282	336	588	300	30	-612	-306	-84	-252	84	66	564	0
YDR146c	276	630	-24	330	1302	564	-258	90	-396	-462	-636	552	2928	-1164	1686	-2190
YLR131c	-210	186	390	660	246	444	-138	306	-384	-1032	-234	828	648	-540	738	-732
YCR005c	-1134	-600	132	-150	-24	534	1608	504	-468	66	828	-156	-1158	-72	-858	156
YCL040w	-3066	-810	372	408	-534	-144	594	402	1854	984	-846	-1212	354	-162	-168	2220
YNR016c	2658	2634	-2022	1698	-4200	-186	-960	-474	4632	-1440	-1062	-1950	1062	-558	1152	-432

4.3.2.2 Studied Models

By considering the presented data above, we construct our model and so the optimization problem. Then after obtaining the result for the corresponding network matrix, we apply our class of numerical schemes including newly introduced ones listed in Eq. (4.26) to generate the approximate time-series gene-expression results and to see the long-term behavior of the system containing selected 26 genes.

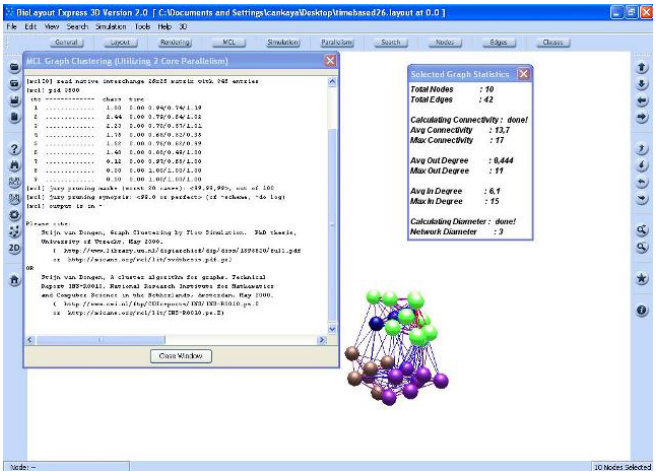
The dynamic network model that we consider for this real-world application is the model $\dot{\mathbf{E}} = \mathbf{ME}$ and the *original* MINLP problem in (4.19), called *(OP-I)*, which is described in details in Subsubsection 4.3.1.1. The constraints in the original MINLP problem are bounded by the maximum indegree/outdegree values $\mathbf{deg_{max}}$ of the considered genes and also by a lower limit $\boldsymbol{\lambda}$ for the degradation rate of the genes. The underlying biological and mathematical motivation for the selection of the bounds and constraints for the restriction of the solution space and so the connections in the network is explained in Section 3.3. By considering this, we can relax and extend our original MINLP problem, *(OP-I)*, by letting the off-diagonal network matrix entries to take negative values and formulate an *extended* MINLP model, called *(OP-II)* only by changing the constraint (4.13) of the original MINLP model in (4.19) in Subsubsection 4.3.1.1 with the constraint (4.14) and formulate a new optimization problem as:

$$\textit{(OP-II)} \qquad \min \quad (4.12), \quad \text{s.t.} \quad \{(4.14), (4.17), (4.18)\}, \tag{4.32}$$

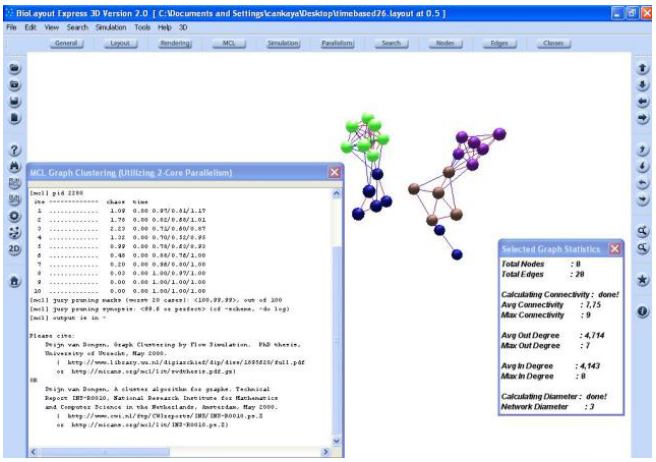
In both presented mixed-integer problems *(OP-I)* and *(OP-II)*, all these bounds can be regarded as realizations of case whose solutions are included in efficiency frontiers (in the case of parametric variation of one bound) or efficiency surface (in the case of parametric variation of several bounds) [79, 82, 139, 140].

For the selection of the corresponding bounds, $\mathbf{deg_{max}}$ and $\boldsymbol{\lambda}$, in the constraints of both original MINLP problem in (4.19) and extended MINLP problem in (4.32), we have used a graphical clustering software *BioLayout Express3D* [154]. This software provides the visualization of the considered network together with some statistics about the network, e.g., connectedness, indegree, outdegree, network diameter values, and so on. A Markov clustering tool is also available in it.

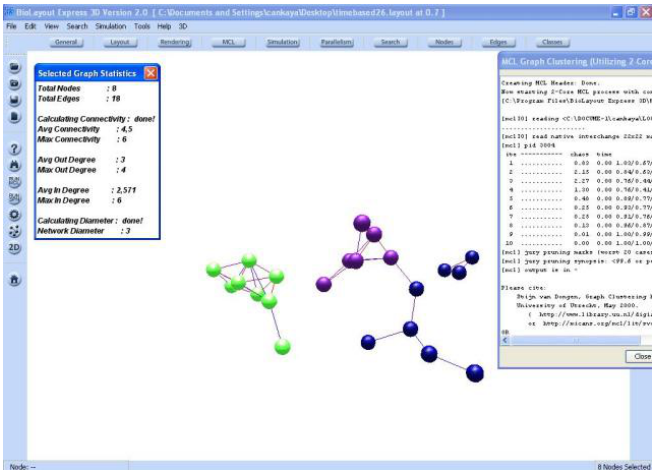
The raw data of our selected 26 genes, given in Table 4.5, are introduced into this program by the corresponding correlation matrix. So, we firstly calculate the correlation matrix corresponding to our gene-wise raw data. Then, we specify a *matrix cut-off value* to define the threshold above which relationships in the network will be shown. After that, we obtained through this software some informations about indegree-outdegree values and also a visualization of our selected small subnetwork as shown in Figure 4.62 for different choices of the matrix cut-off values.



(a)



(b)



(c)

Figure 4.62: By using BioLayout Express3D software, indegree and outdegree analysis of selected 26 genes by the corresponding correlation matrix having (a) cut-off value= 0, (b) cut-off value= 0.5 and (c) cut-off value= 0.7

In order to choose the matrix cut-off value, we briefly look at the correlation matrix, its inverse and covariance matrices of our raw gene data. Then, we select the possible cut-off values as 0.5, 0.7, 0.8 in order not to loose some nodes (genes) of the network but also to eliminate some weak relations (edges) among the genes. According to the obtained statistical results from that software, we took the corresponding maximum indegree value, that can be common for all genes in our subnetwork, as $deg_{max,i} = 6$, for $i = 1, 2, \ldots, 26$.

As the next bound for our network, we have to decide the a lower limit of the degradation rate λ corresponding to our subnetwork. For this purpose, we perform *log-like linear regression* analysis for each gene in our subnetwork. Then, we look at the *slope* of each regression line that is obtained in log_2-scale and select the average value as our common limit for the degradation rate. Hence, we specify $\lambda(i) = 12$ for $i = 1, 2, \ldots, 26$.

Therefore, after all these steps and analysis which can be called as preprocessing, we decide to take the bounds in the constraints of the original and extended MINLP problems as follows

$$\lambda(i) = 12, \quad deg_{max,i} = 6 \qquad (i = 1, 2, \ldots, 26). \tag{4.33}$$

We are now be able to study both original and extended MINLP problems for our network with the selected bounds of the constraints in (4.33) and then apply all our numerical schemes listed in Eq. (4.26) for the gene data in Table 4.5 and derivative data in Table 4.6 to obtain time-series predictions.

Used software: For solving the original MINLP model *(OP-I)* and extended MINLP model *(OP-II)*, we have used the *IBM ILOG CPlex* Optimizer and the *Gurobi* Optimizer tools in order to prove global optimality.

4.3.2.3 Numerical Results

By considering the data given in Table 4.5 and Table 4.6, we calculated the following network matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ as the solutions of the studied *original* MINLP model and *extended* MINLP model, respectively. The calculated matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ have both 6 real and 20 complex (distinct) eigenvalues being both negative and positive.

After we obtained the network matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ as the solutions of two optimization problems, we apply all four numerical methods presented in Eq. (4.26) to generate approximate

gene-expression values. Comparison of the approximate results produced by all these numerical schemes is done for the original MINLP model. Additionally, we compare the results of original MINLP model and extended MINLP model by fixing the used numerical scheme as Euler's method. In both comparisons, the step-size is taken as $h_k = 1/6$ (hr). For each gene of our subnetwork, all corresponding results are presented in the following graphs in part *(i)* and *(ii)*.

$$\mathbf{M}_1 = \begin{pmatrix}
-7.36 & 0 & 11.36 & 0.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.50 & 0.57 & 0 & 0 \\
0 & -8.69 & 2.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.77 & 0 & 0 & 0 & 0 & 2.52 & 0 & 0 & 0 & 0.48 & 0 & 0 & 0.49 & 0 \\
0 & 0 & -4.08 & 2.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.59 & 0 & 0 & 0 & 0.045 & 0 & 0 & 0 & 0.22 & 0 \\
0 & 0 & 0 & -4.73 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.42 & 0.92 & 0.06 & 0 & 0 & 0 & 0.15 & 0.35 & 0 \\
0 & 8.55 & 0 & 0 & -4.80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 24.39 & 2.38 & 0 & 12.09 & -11.16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.11 \\
0 & 15.40 & 0 & 0 & 5.89 & 0 & -5.07 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.58 & 0 \\
4.98 & 12.32 & 0 & 0 & 0 & 1.77 & 0 & -8.91 & 0 & 37.82 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.07 \\
0 & 5.33 & 0 & 0 & 2.60 & 0 & 0 & 0 & -5.52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.06 & 0 \\
0 & 0.37 & 0 & 0 & 0.63 & 0 & 0 & 0 & 0.028 & -5.17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 \\
1.26 & 6.22 & 4.56 & 0 & 4.37 & 0 & 0 & 0 & 3.37 & 0 & -12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.73 & 0 & -3.51 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.21 \\
0 & 0.26 & 0 & 0 & 0 & 0 & 0.07 & 0 & 0 & 2.15 & 0 & 0.65 & -6.28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.14 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.88 & 0 & 0 & 6.10 & 0 & 36.68 & -8.13 & 1.59 & 0 & 0 & 0 & 0 & 0 & 0 & 4.87 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.95 & 0 & 56.81 & 4.51 & 0 & 0 & 0 & -5.99 & 0 & 0 & 0 & 0 & 0 & 0 & 2.96 & 0 & 1.39 & 0 & 0 \\
0 & 2.34 & 0 & 1.30 & 0 & 0 & 0 & 0 & 1.28 & 0 & 0 & 6.93 & 0 & 0 & 0.43 & -6.57 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.66 & 0 & 0 & 0 & 0 & -3.37 & 0.07 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.62 & 0 & 0 & 0 & 0 & 4.46 & -5.51 & 0 & 0 & 0 & 0.98 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.31 & 0 & 0 & 0 & 0 & 7.63 & 4.38 & -7.87 & 0.03 & 0 & 0 & 0.47 & 0 & 0 & 0 \\
2.25 & 0 & 23.76 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.89 & 0 & 0 & 0 & 0 & 10.66 & 0 & 0 & -10.18 & 0 & 0 & 12.29 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.07 & 0 & 0 & 0 & 0.61 & 0 & 0 & 0 & 0 & 0 & 0 & 3.78 & 0 & -6.05 & 0 & 0 & 0 & 0.26 & 0.06 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.62 & 0 & 0.04 & 0.32 & 0 & 0 & 4.49 & 0 & 0 & 0 & -4.51 & 0 & 1.35 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.68 & 0 & 0 & 0.28 & 0 & 0 & 1.15 & 3.07 & 0 & 0 & 0 & -6.26 & 0.39 & 0 & 0 \\
0.09 & 0 & 0 & 0 & 0 & 0 & 0 & 0.23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.94 & 0 & 0 & -2.89 & 0 & 0 \\
1.52 & 0 & 0 & 2.17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.60 & 0 & 0.29 & -4.84 & 0 \\
0 & 7.05 & 8.95 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 35.91 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.63 & -4.85
\end{pmatrix} \tag{4.34}$$

$$
\mathbf{M}_2 = \begin{pmatrix}
-6.62 & 0 & 23.26 & 0 & 0 & 0 & -1.52 & 0 & 0 & 0 & 0 & 0 & 1.13 & 0 & 0 & 0 & 0 & -1.52 & 0 & 0 & 0 & 0 & -1.33 & 0 \\
-1.19 & -9.02 & 6.01 & 0 & -1.36 & 0 & 0 & 0 & 0 & 0 & 5.52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.80 & 0 \\
-1.20 & 0 & 0 & 0 & 0 & 0 & 0 & -1.41 & 0 & 0 & 0 & 0 & 0 & 0 & 0.82 & 0.16 & 0 & 0 & 0 & 0 & 0 & 0.76 & 0 & 0.41 \\
0 & 0 & 0 & -5.76 & 3.29 & 0 & 0 & -3.74 & 0 & 0 & 0 & 0 & 0 & 0 & 5.47 & 0 & 0 & 0 & 0 & 0 & 0 & 1.44 & 0 & 0.31 \\
-1.56 & 16.68 & 0 & 1.24 & 0 & 0 & 0 & 0 & 0 & -4.93 & 0 & 0 & 0 & 0 & -4.22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.66 & 0 \\
0 & 35.76 & 0 & 6.28 & 0 & -4.06 & 0 & -2.67 & 0 & 0 & 13.62 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -11.68 & 0 & 0 & 0 & 0 \\
0 & 0 & 72.79 & 0 & 0 & -8.87 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 92.25 & 0 & 0 & 0 & -79.09 & 0 & 0 & -12.72 & 8.69 & 0 \\
0 & 0 & 30.35 & 0 & 31.10 & 0 & -3.49 & 0 & 0 & -32.77 & 0 & 0 & 0 & 15.37 & 0 & -18.52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 6.52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.81 & 0 & 0 & -0.83 & 0 & 0 & 0 & 0 & 0 & -1.11 & 0 & 0 & 3.22 & 0.63 \\
0 & 0.58 & 0 & 0 & 0 & 0 & 0 & 0 & -2.74 & 0 & 0 & 0.32 & 0 & 0 & -1.03 & 0 & 0 & 0 & 0 & 0 & 0 & 0.24 & 0.27 \\
0 & 5.76 & 14.52 & 0 & 0 & 0 & 0 & 3.89 & 0 & -9.78 & 0 & 0 & 0 & 0 & 5.28 & 0 & 0 & -10.08 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.76 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & -6.39 & 0 & 0 & 0 & -2.43 & 0 & 1.90 & 0 & 0 & 0 & 0 & 0.40 \\
0 & 0 & 1.67 & 0 & 0 & 0 & 0.10 & 0 & 0 & -0.69 & 0 & 0 & 0.12 & 0 & 0 & 0 & 0 & 0 & 0 & -1.28 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.10 & 0 & 0 & 0 & 51.21 & -7.40 & 0 & 0 & 0 & 0 & -15.48 & 7.63 & 0 & 5.65 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.80 & 0 & 0 & 0 & 34.32 & 0 & -5.32 & 0 & 0 & 0 & 0 & 0 & 0 & 9.14 & -3.29 & 0 \\
0 & 3.82 & 0 & 0 & 1.20 & -0.57 & 0 & 0 & 0 & 7.49 & 0 & 0 & -0.29 & 0 & 0 & 0 & 0 & 0 & 0 & -1.07 & 0 & 0 & 0 \\
1.16 & 0 & -2.73 & 0 & 0 & 0 & 0.69 & 0 & -1.40 & 0 & 0 & 0 & -0.055 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.14 & 0 & 0 \\
0 & 0 & -4.10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.56 & 0 & 3.75 & 0 & 0.38 & 0 & 0 & 0 & 0 & 0.65 & 0 \\
0 & -2.92 & 0 & 0 & 0 & -0.94 & 0 & 0 & 0 & -10.54 & 2.37 & 8.40 & 0 & 0 & 0 & -6.89 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 30.88 & 22.42 & 0 & 0 & 0 & 0 & 0 & -22.04 & 0 & 13.71 & 0 & 0 & 0 & 13.80 & 0 & 0 & -11.94 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.33 & -0.86 & 0 & 0 & 2.13 & 0 & 0 & 0 & 0 & -4.42 & 0 & 3.87 & 0 & -4.75 & 0 & 0 & 0 \\
0 & 0 & 0 & 13.00 & 0 & 0 & 0 & 0 & 0 & 0 & 21.82 & 0 & 1.29 & 0 & 0 & 0 & 0 & 0 & -10.18 & -5.21 & 0 & 0 & -3.08 \\
0 & 0 & 0 & 2.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6.78 & 0 & 3.26 & 0 & 0 & -2.09 & -5.08 & 0 & -1.12 \\
0 & 0 & -6.76 & 0 & -3.95 & 0 & 0 & 0 & 0 & 6.53 & 0 & 0 & 0 & 0 & -3.56 & 0 & 0 & 0 & 11.63 & 0 & 0 & -2.66 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 17.34 & 0 & -9.72 & 0 & 0 & 0 & 0 & -11.75 & 0 & 0 & -0.34 & 0 & 11.95 & 0 & -3.61 \\
0 & 0 & 31.32 & 0 & 0 & -4.30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.64 & 15.96 & 0 & 0 & 0 & -3.10 & -17.89 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{4.35}
$$

(i) Comparison of the results of all numerical methods for $h_k = 1/6$ with original model

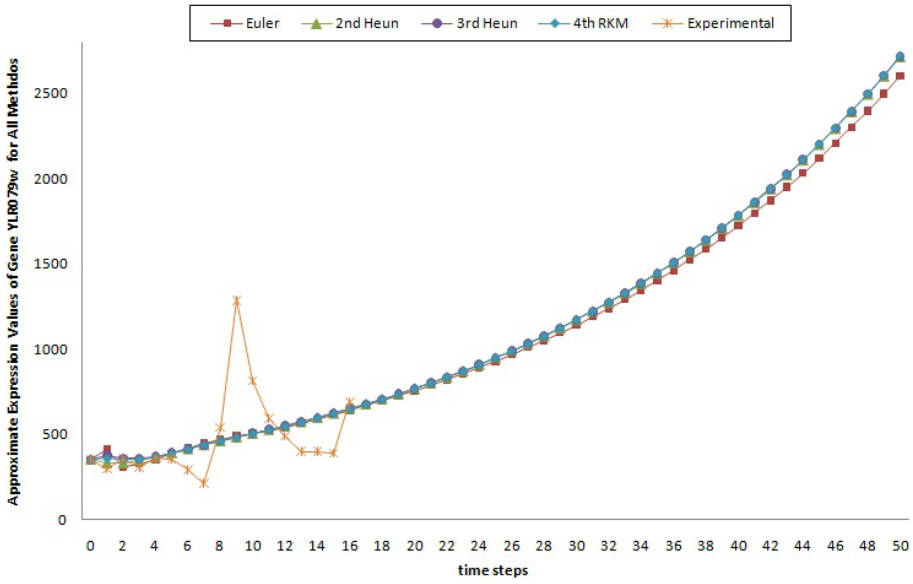


Figure 4.63: Results of all 4 schemes for Gene $YLR079w$ by considering original model

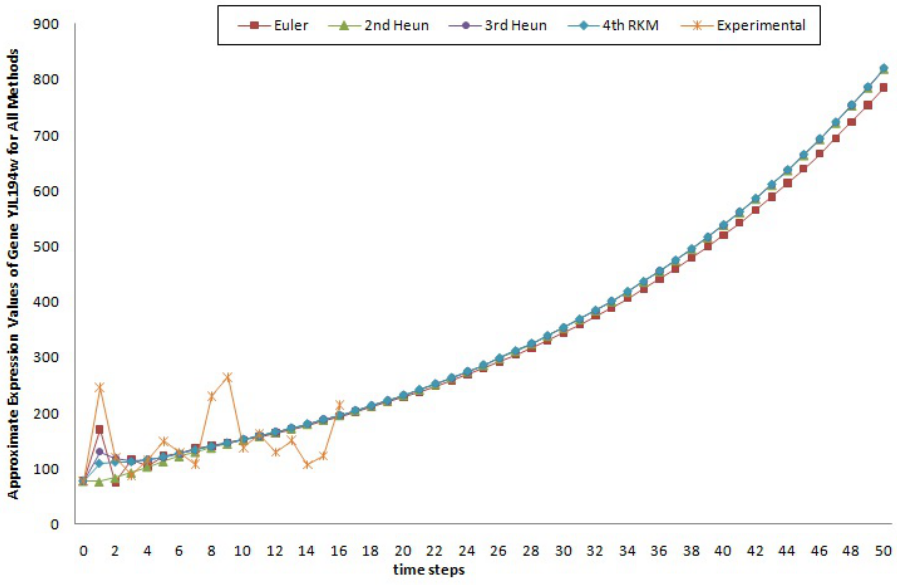


Figure 4.64: Results of all 4 schemes for Gene $YJL194w$ by considering original model

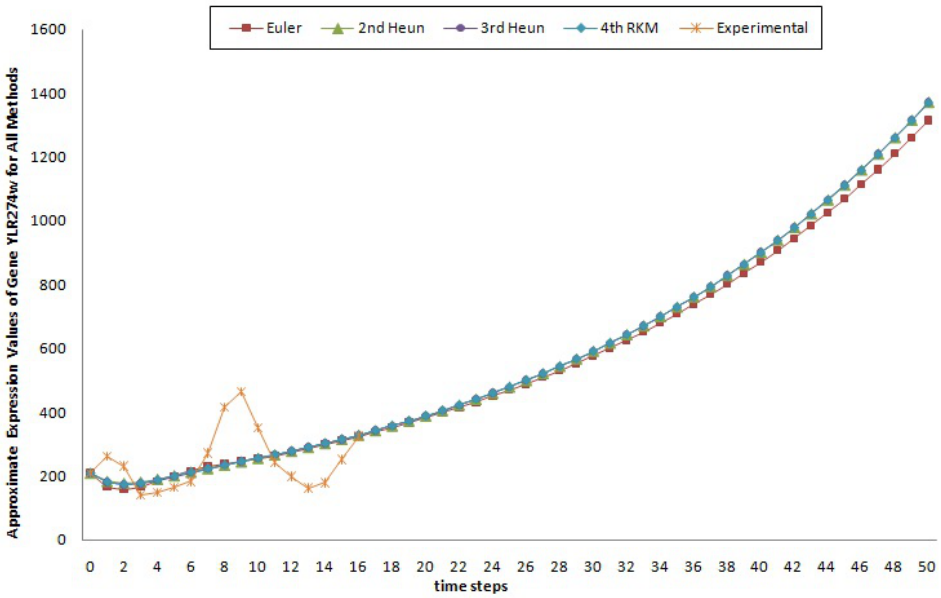


Figure 4.65: Results of all 4 schemes for Gene *YLR*274*w* by considering original model

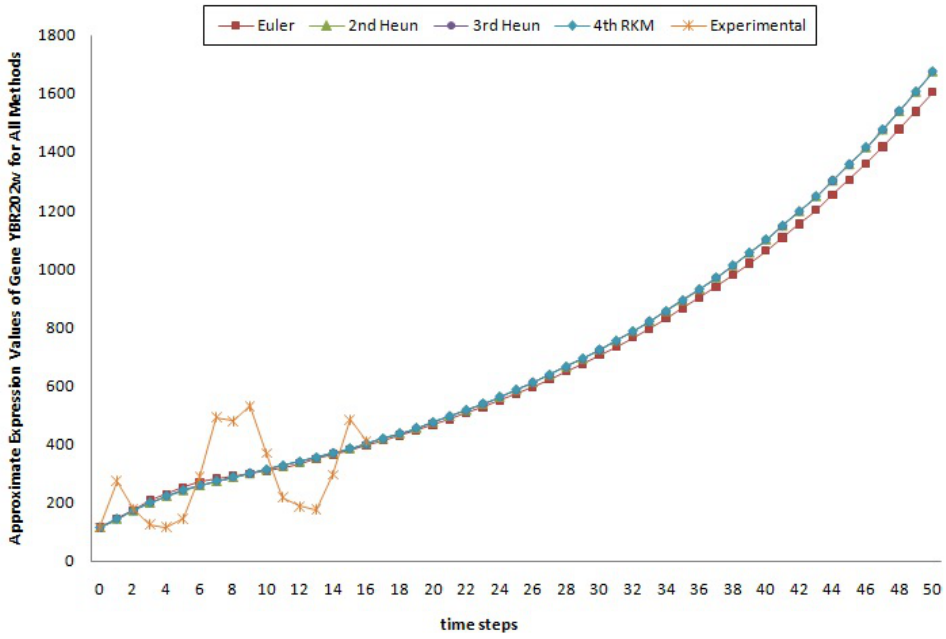


Figure 4.66: Results of all 4 schemes for Gene *YBR*202*w* by considering original model

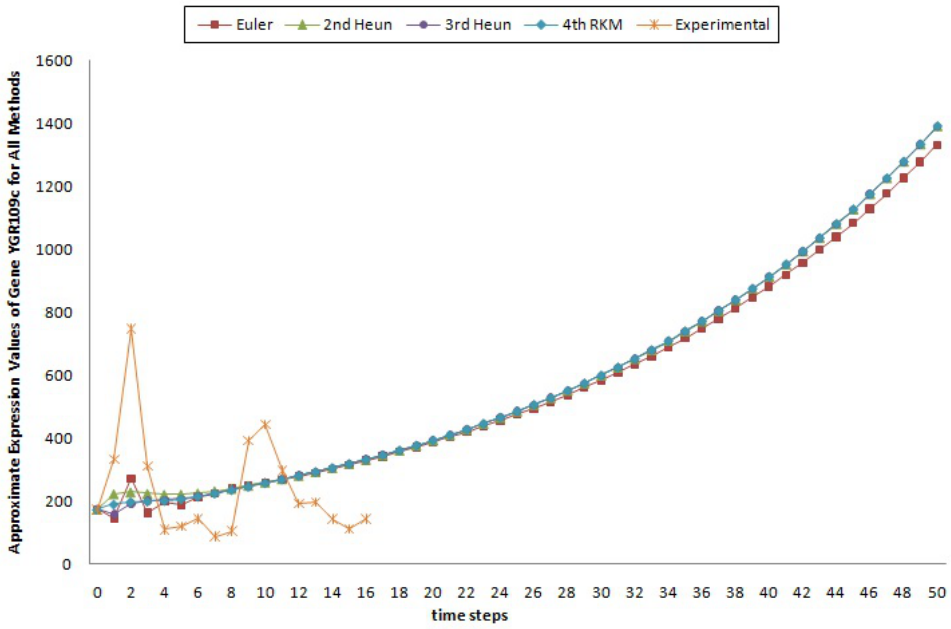


Figure 4.67: Results of all 4 schemes for Gene $YGR109c$ by considering original model

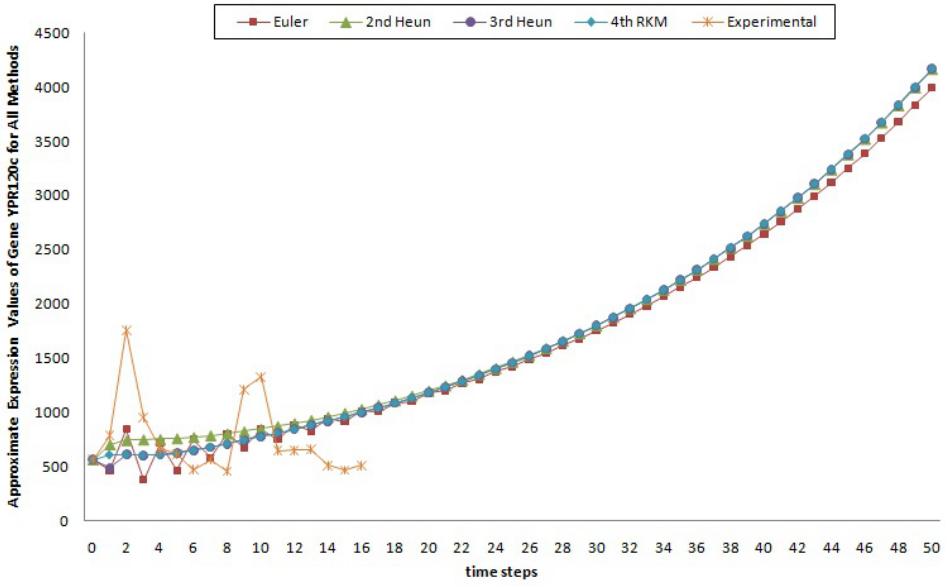


Figure 4.68: Results of all 4 schemes for Gene $YPR120c$ by considering original model

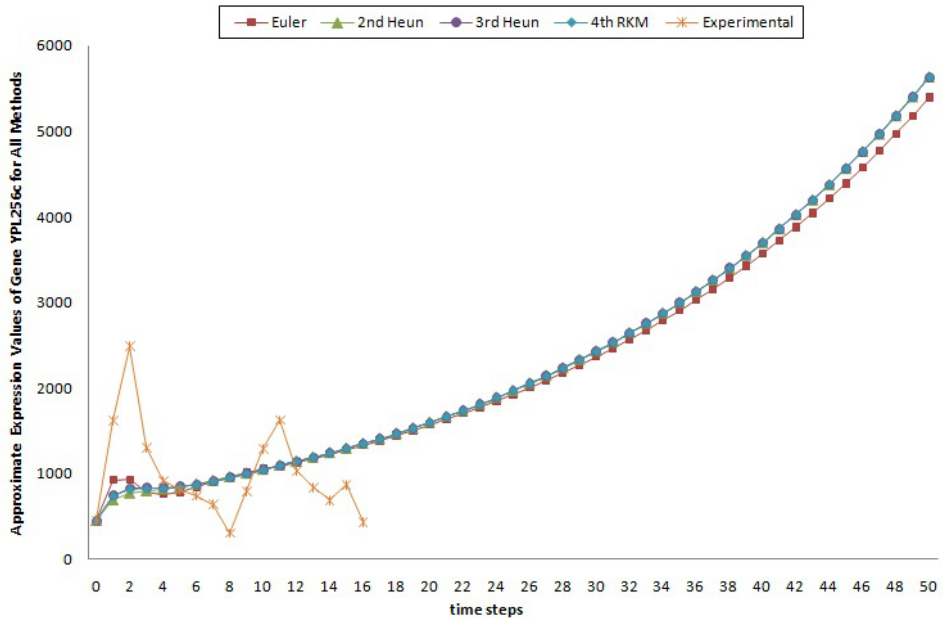


Figure 4.69: Results of all 4 schemes for Gene $YPL256c$ by considering original model

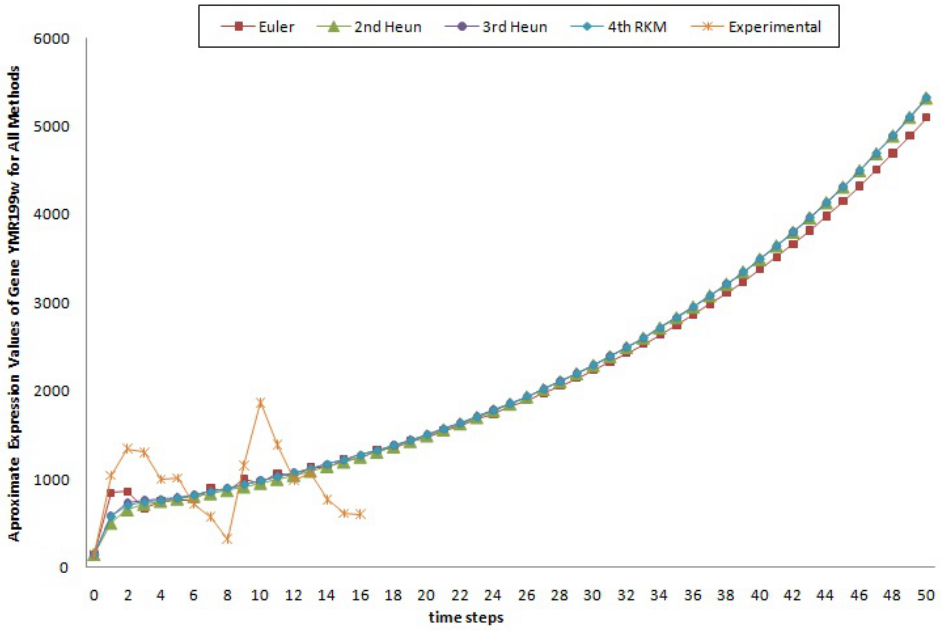


Figure 4.70: Results of all 4 schemes for Gene $YMR199w$ by considering original model

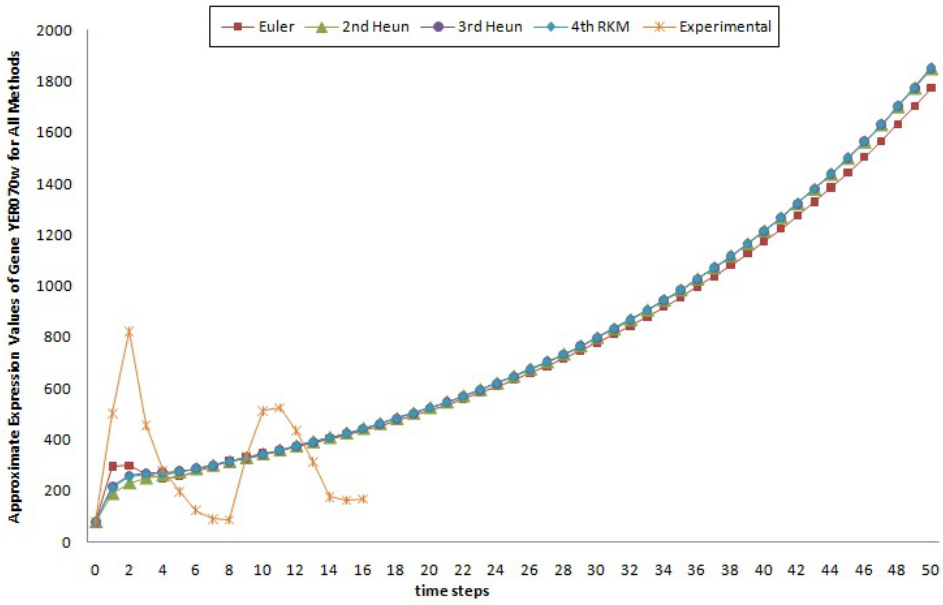


Figure 4.71: Results of all 4 schemes for Gene $YER070w$ by considering original model

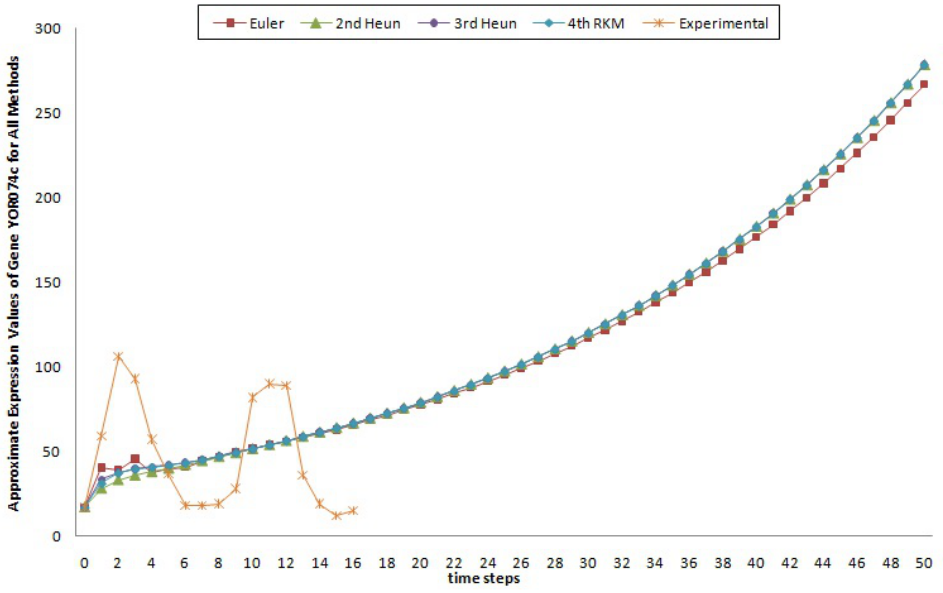


Figure 4.72: Results of all 4 schemes for Gene $YOR074c$ by considering original model

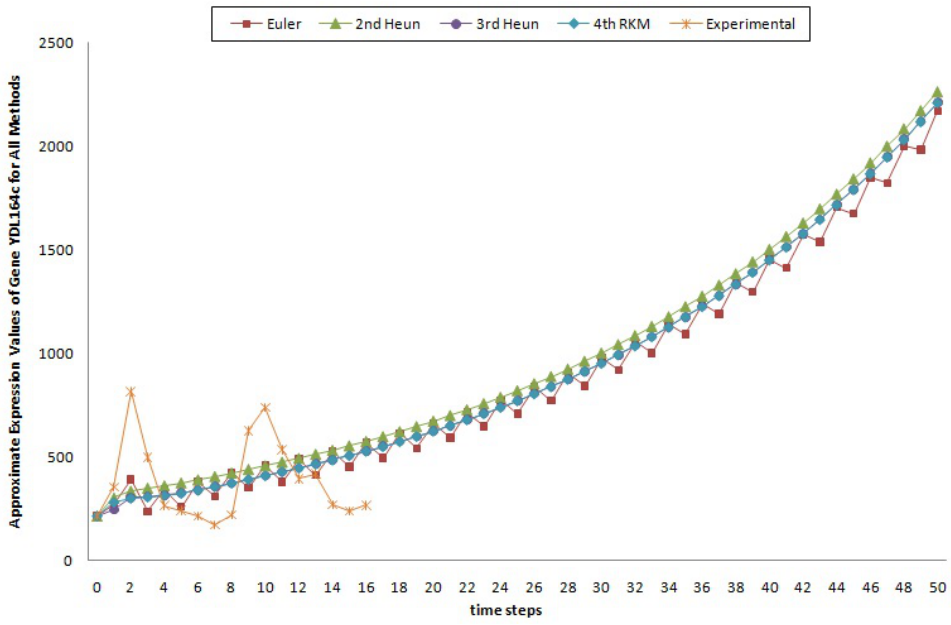


Figure 4.73: Results of all 4 schemes for Gene $YDL164c$ by considering original model

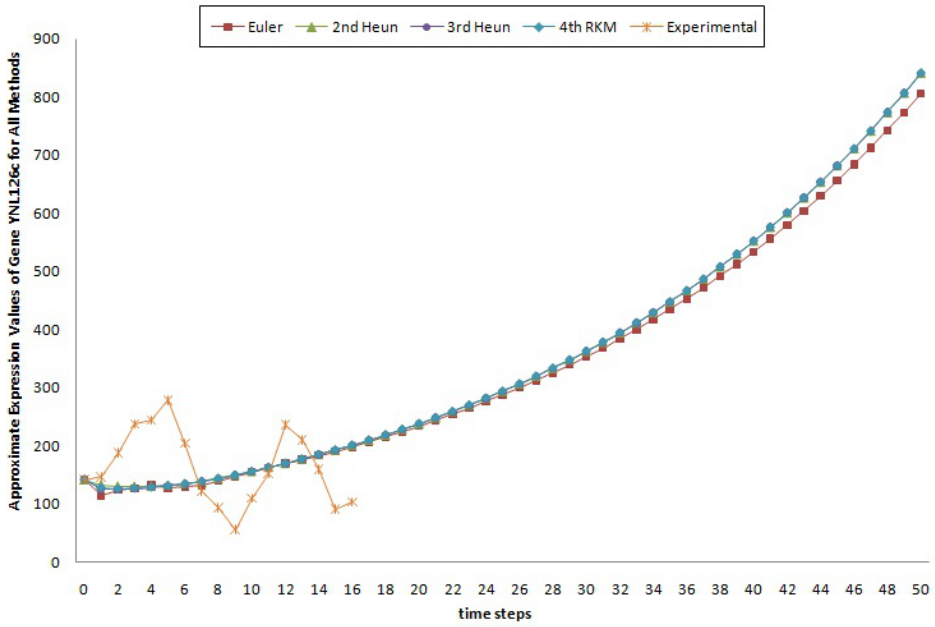


Figure 4.74: Results of all 4 schemes for Gene $YNL126c$ by considering original model

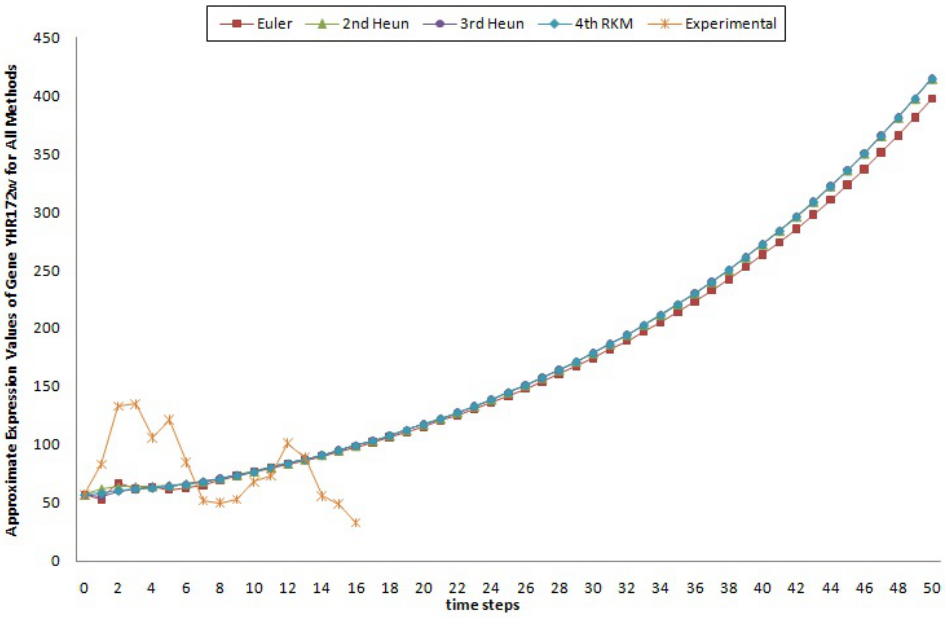


Figure 4.75: Results of all 4 schemes for Gene $YHR172w$ by considering original model

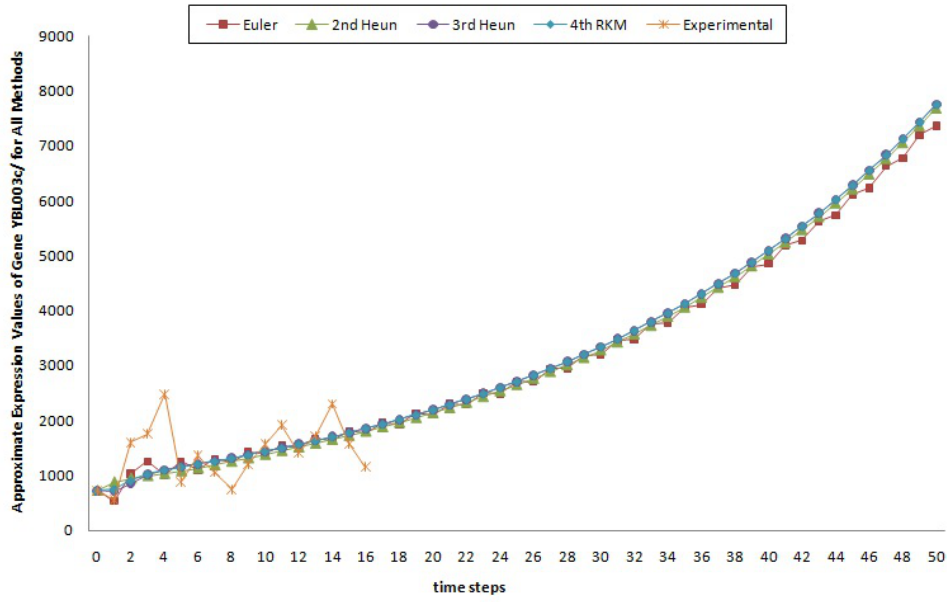


Figure 4.76: Results of all 4 schemes for Gene $YBL003c$ by considering original model

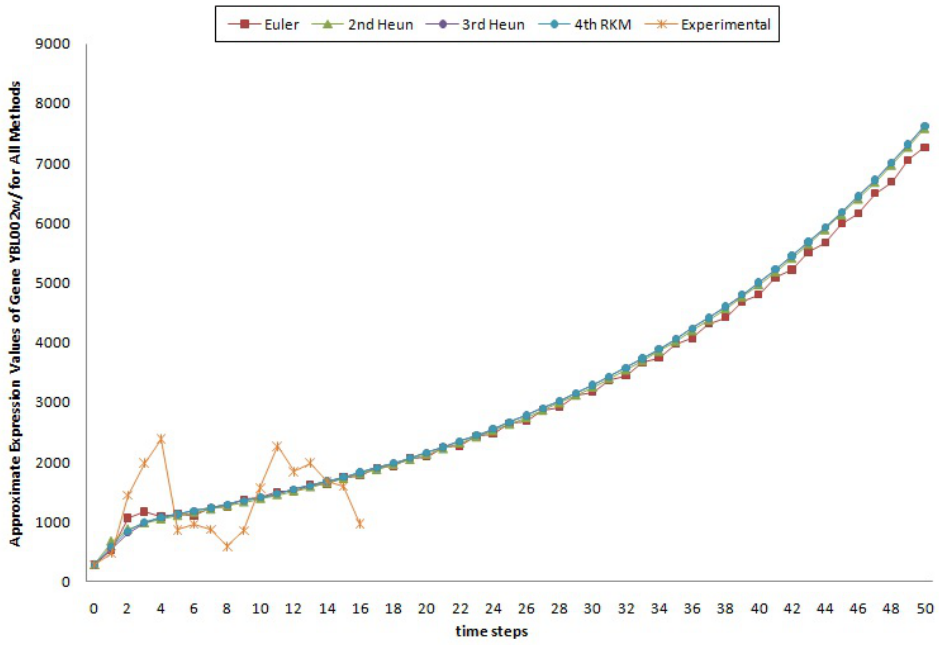


Figure 4.77: Results of all 4 schemes for Gene $YBR002w$ by considering original model

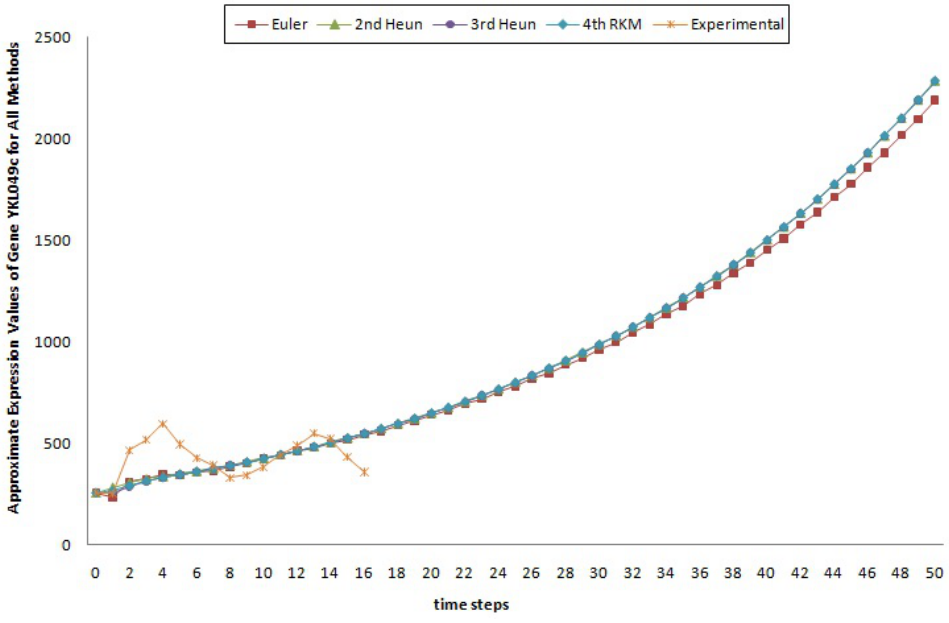


Figure 4.78: Results of all 4 schemes for Gene $YKL049c$ by considering original model

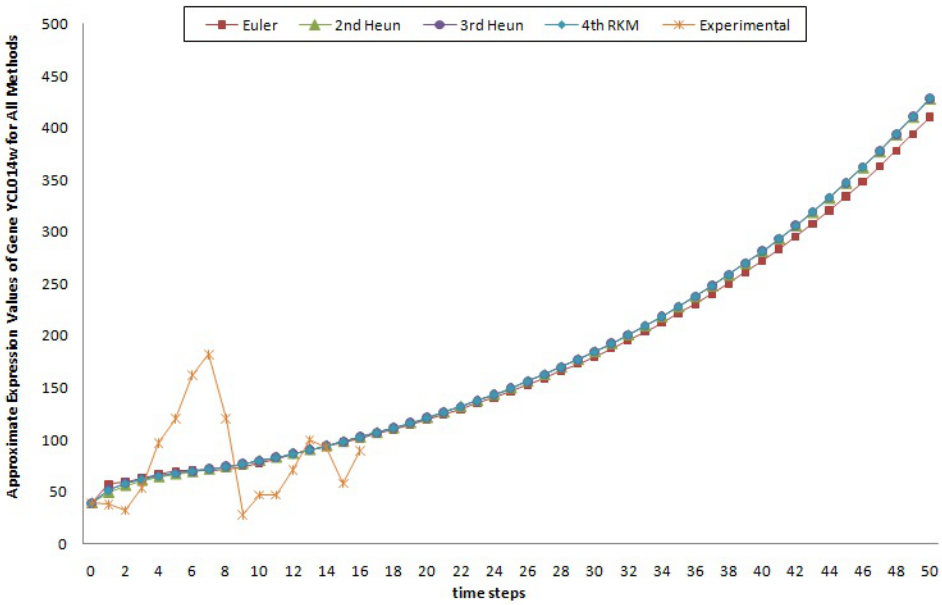


Figure 4.79: Results of all 4 schemes for Gene $YCL014w$ by considering original model

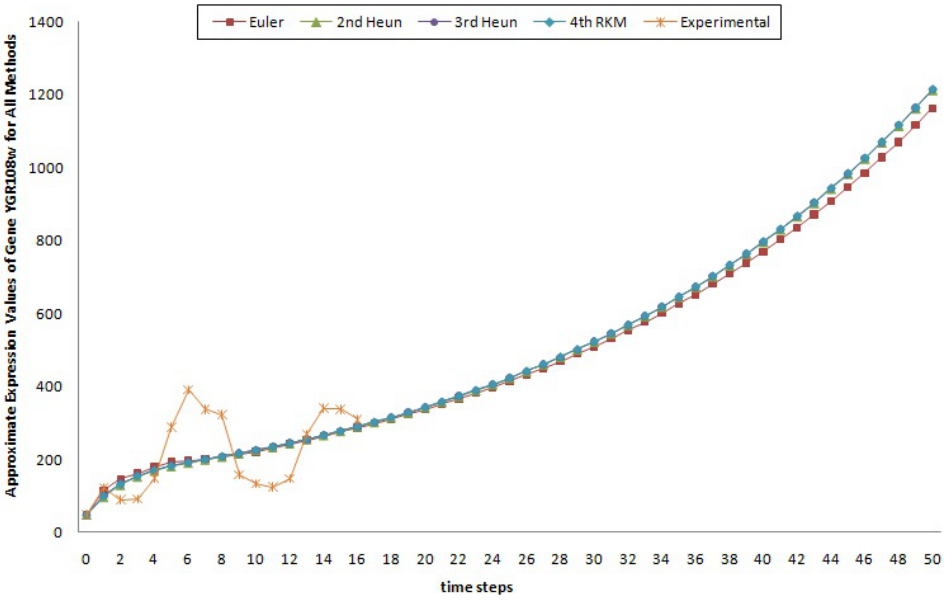


Figure 4.80: Results of all 4 schemes for Gene $YGR108w$ by considering original model

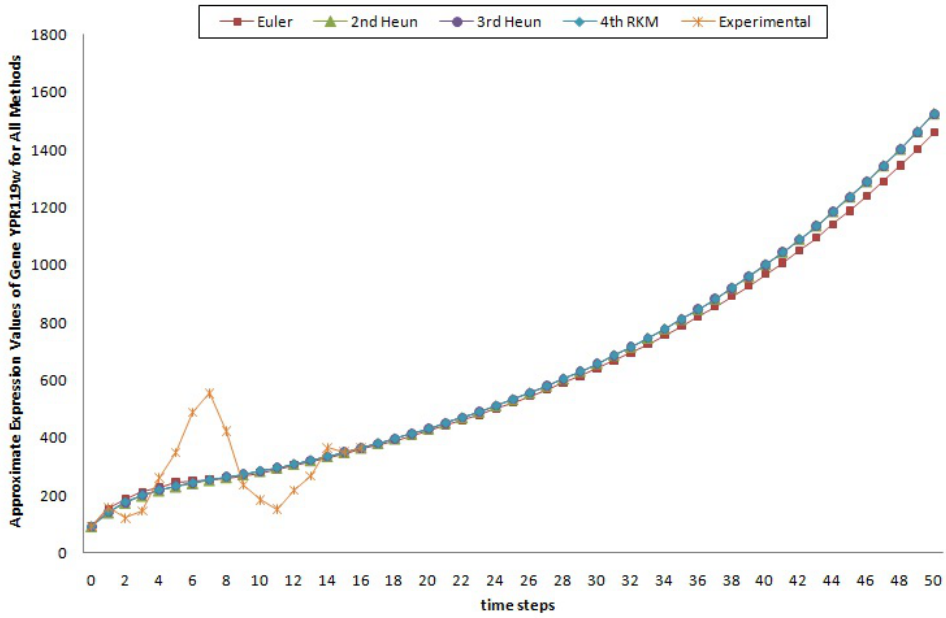


Figure 4.81: Results of all 4 schemes for Gene $YPR119w$ by considering original model

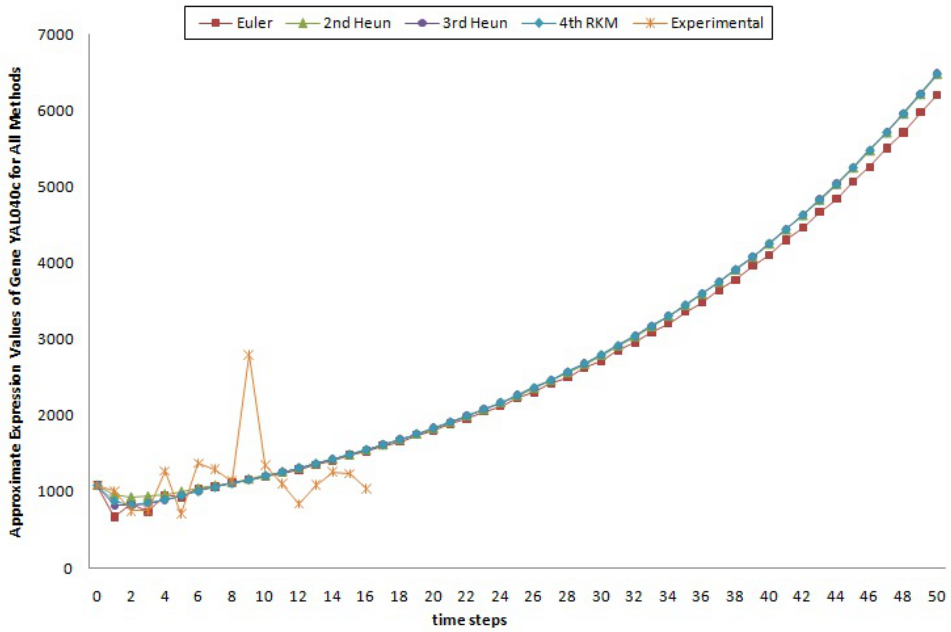


Figure 4.82: Results of all 4 schemes for Gene $YAL040c$ by considering original model

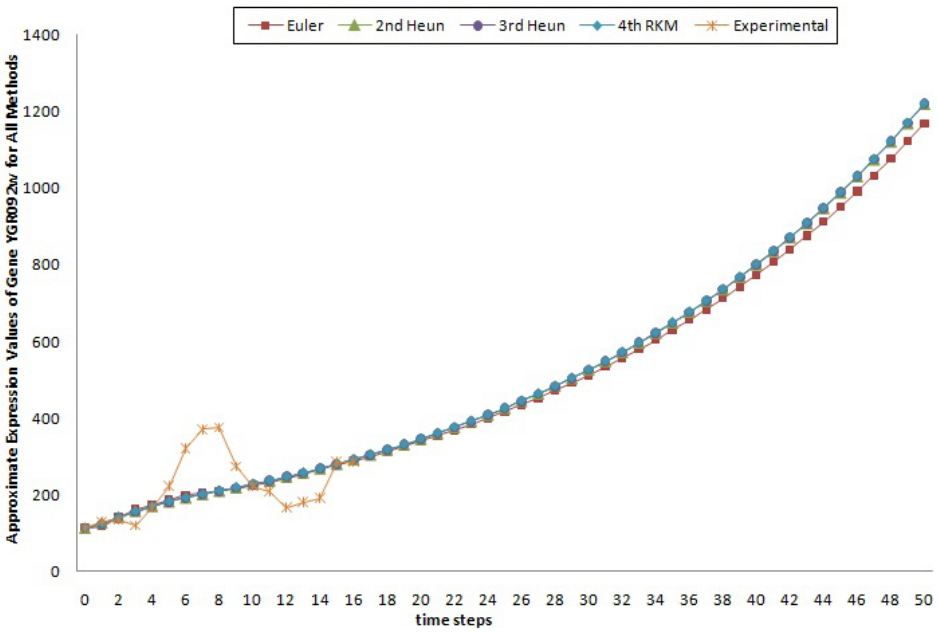


Figure 4.83: Results of all 4 schemes for Gene *YGR*092*w* by considering original model

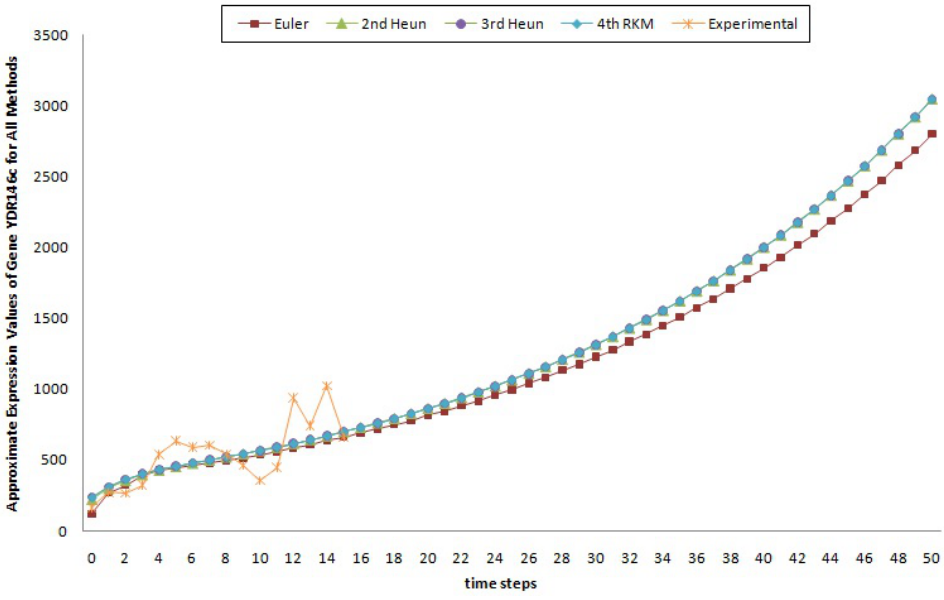


Figure 4.84: Results of all 4 schemes for Gene *YDR*146*c* by considering original model

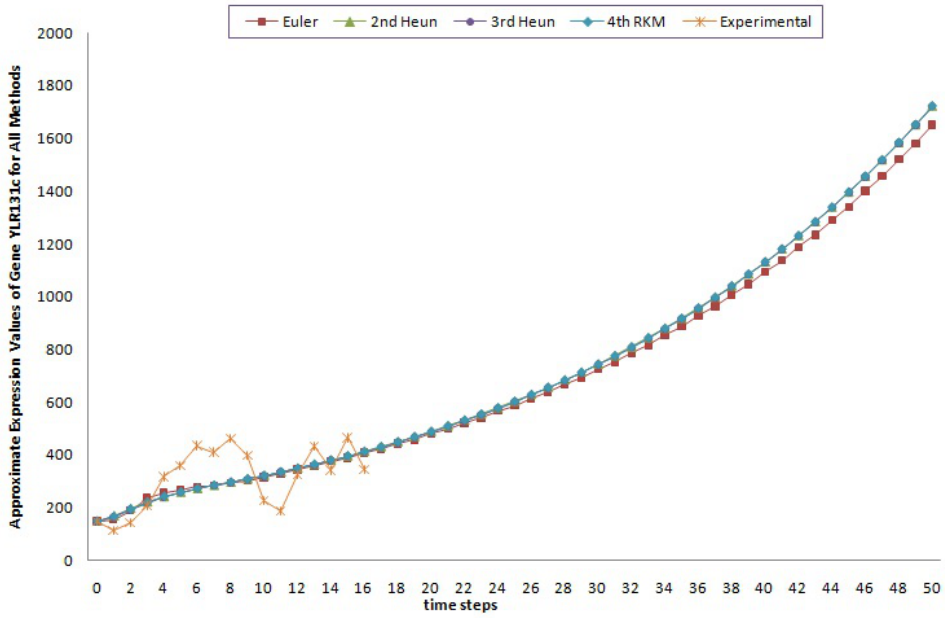


Figure 4.85: Results of all 4 schemes for Gene $YLR131c$ by considering original model

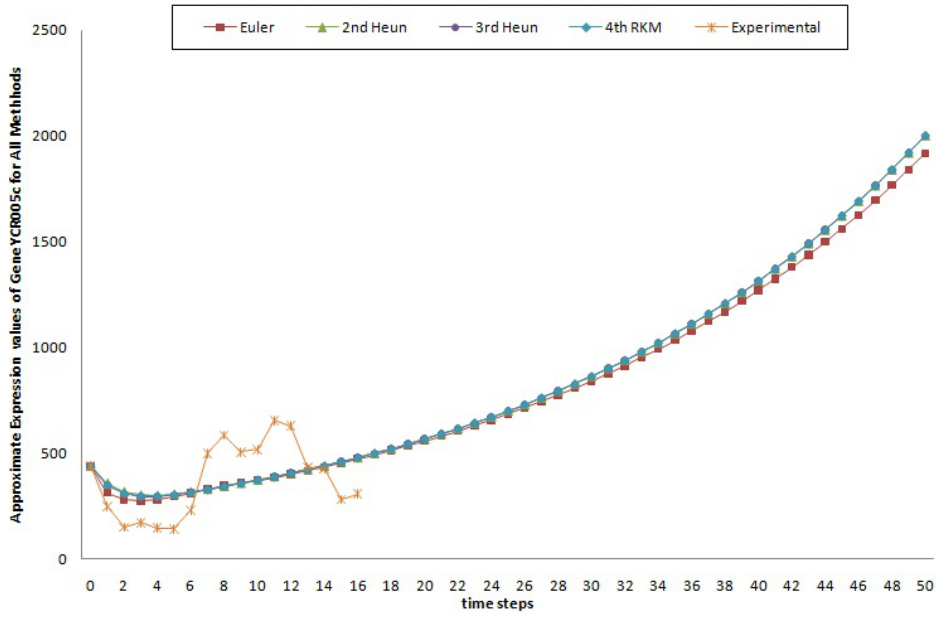


Figure 4.86: Results of all 4 schemes for Gene $YCR005c$ by considering original model

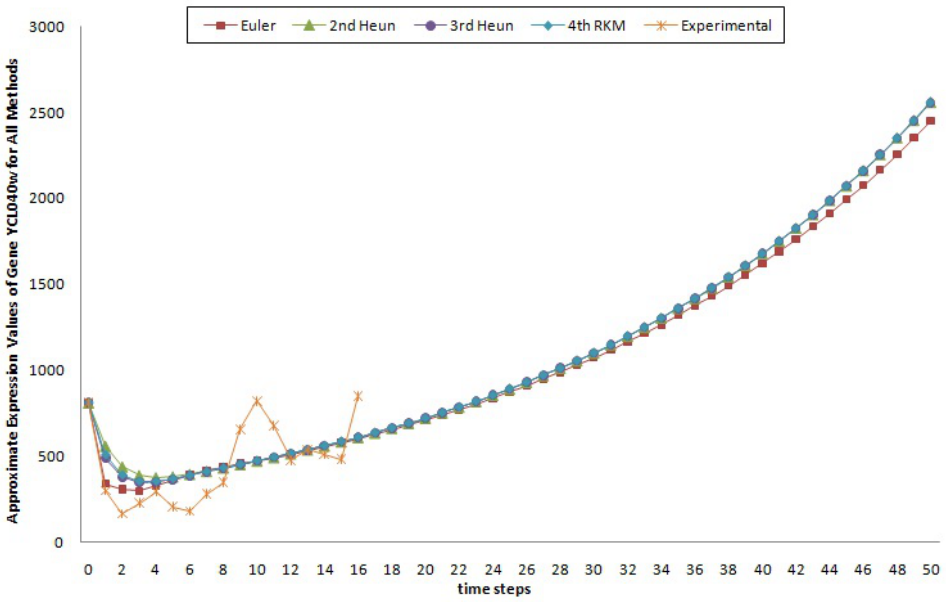


Figure 4.87: Results of all 4 schemes for Gene $YCL040w$ by considering original model

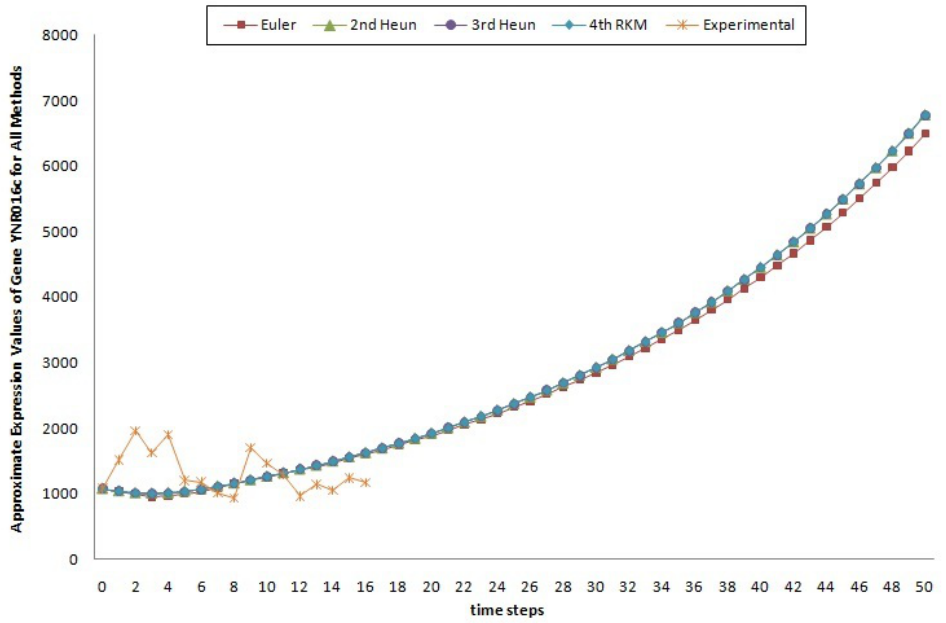


Figure 4.88: Results of all 4 schemes for Gene $YNR016c$ by considering original model

(ii) Comparison of the results of original and extended models with Euler method for $h_k = 1/6$ *presented for some of the genes*

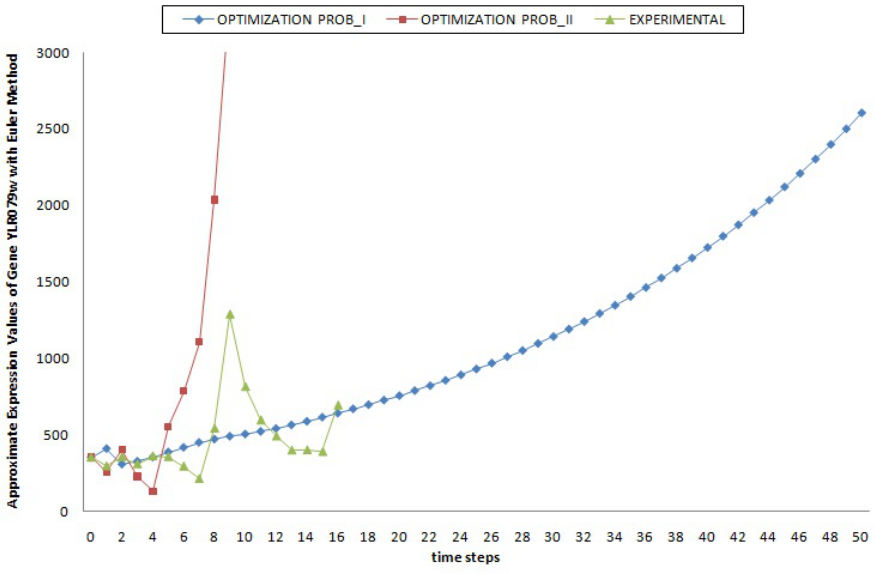


Figure 4.89: Results of Gene *YLR*079*w* obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

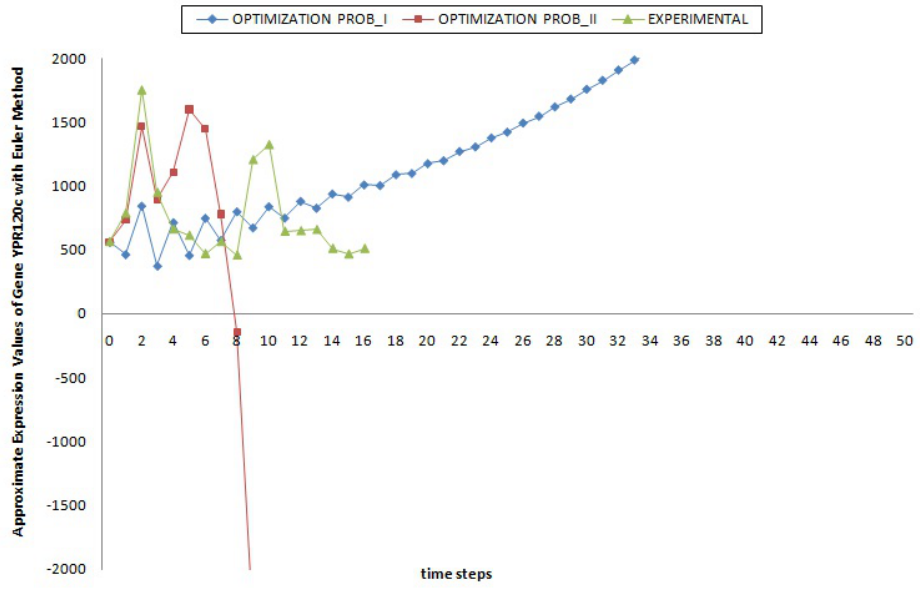


Figure 4.90: Results of Gene $YPR120c$ obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

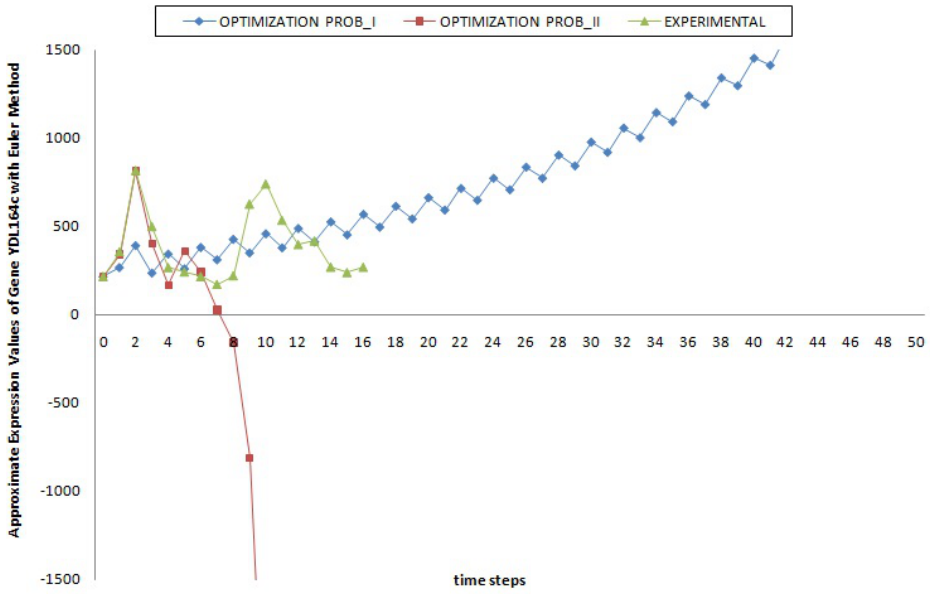


Figure 4.91: Results of Gene $YDR164c$ obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

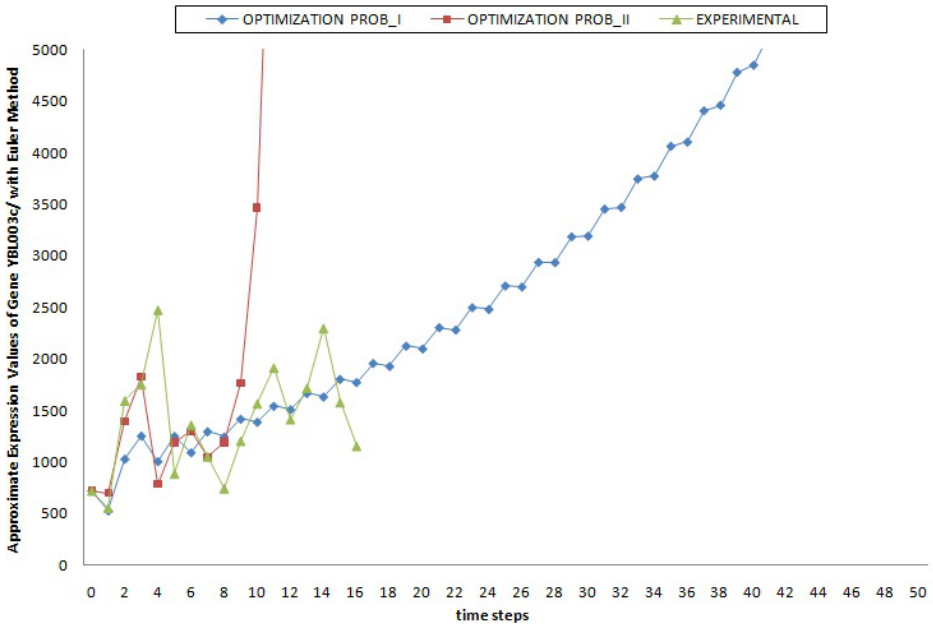


Figure 4.92: Results of Gene $YBL003c$ obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

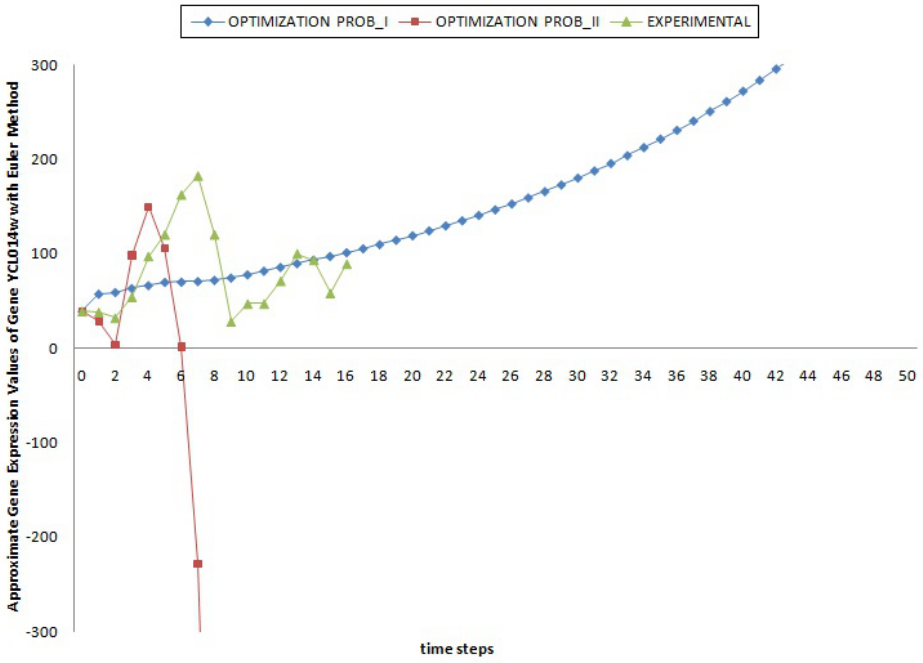


Figure 4.93: Results of Gene $YCL014w$ obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

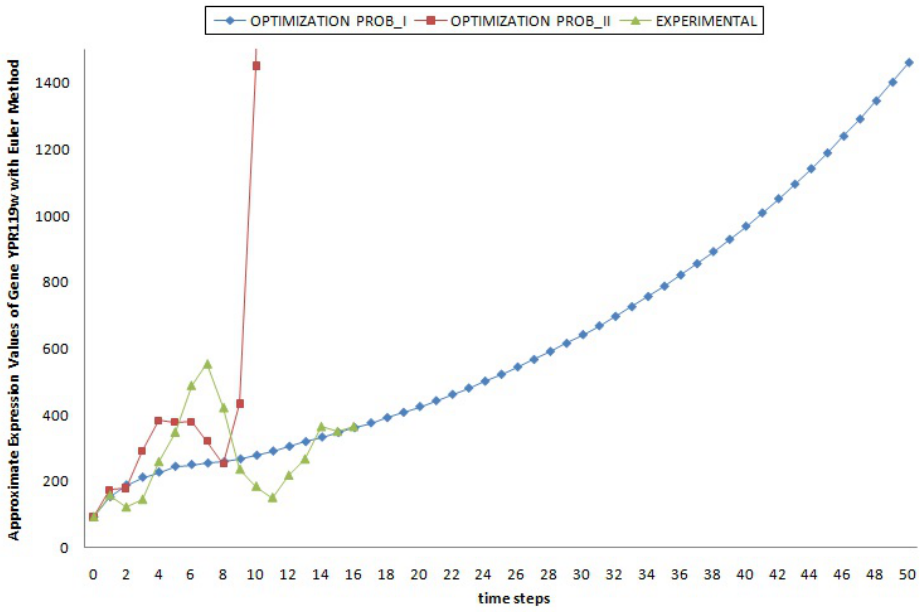


Figure 4.94: Results of Gene $YPR119w$ obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

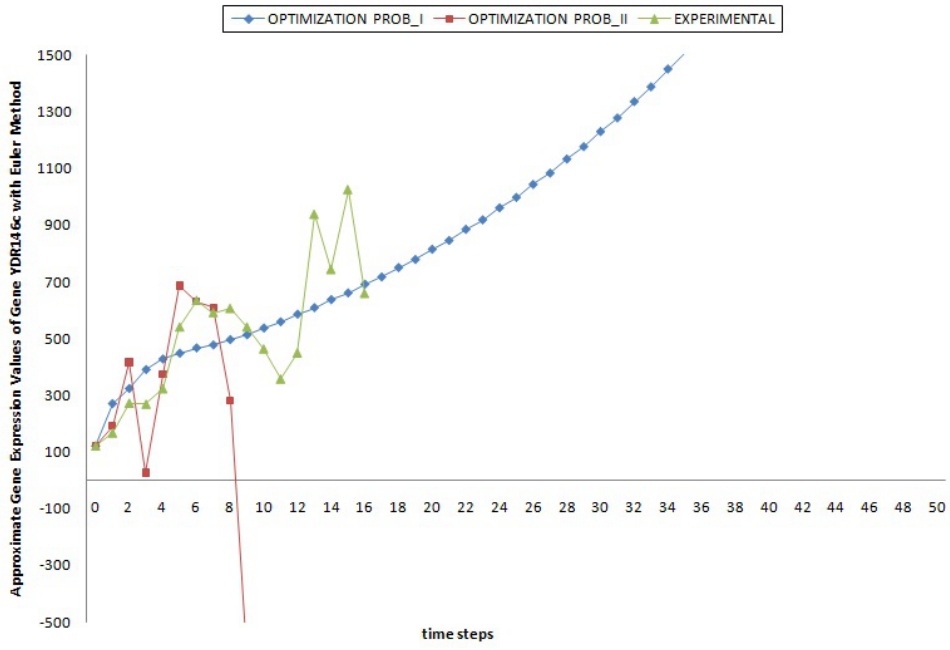


Figure 4.95: Results of Gene $YDR146c$ obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

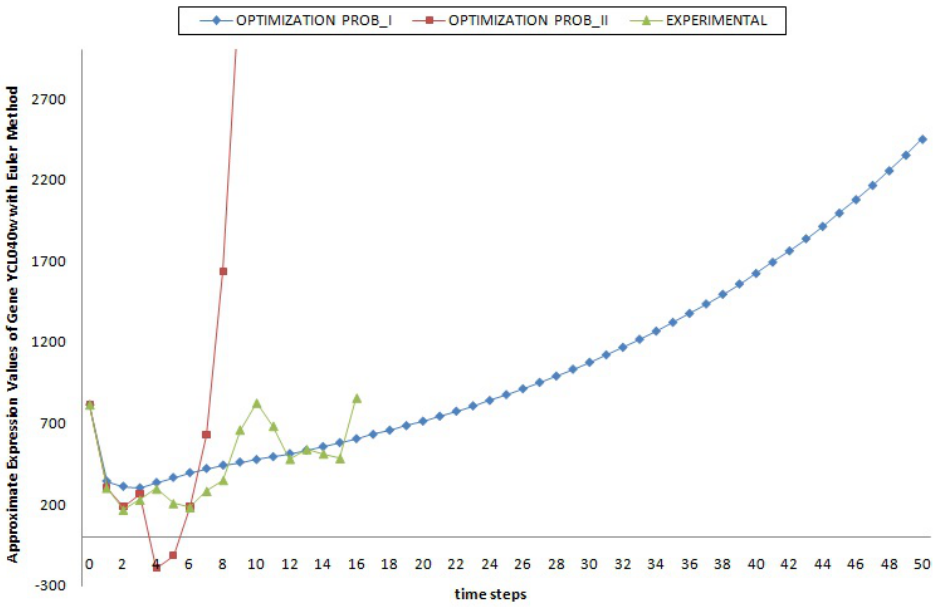


Figure 4.96: Results of Gene $YCL040w$ obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

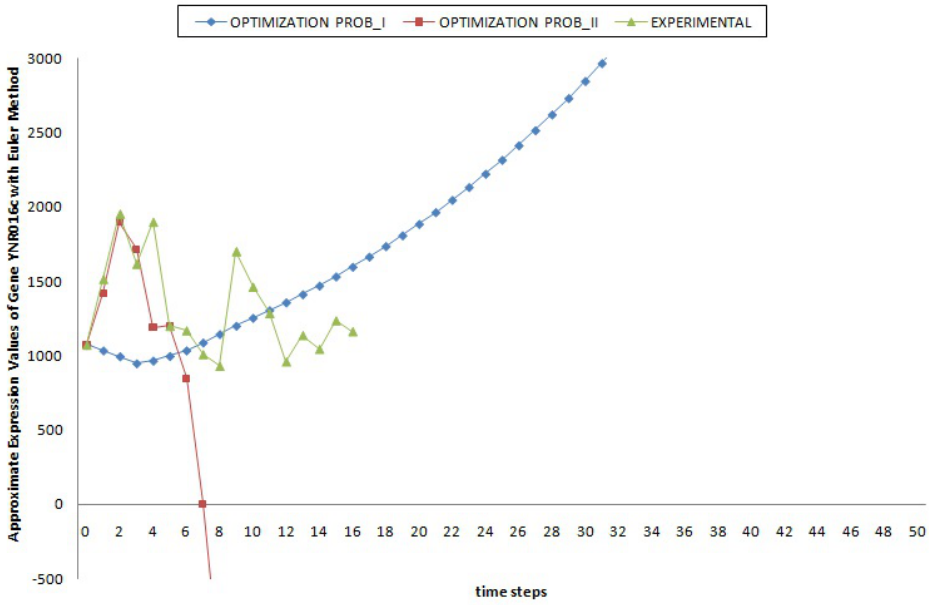


Figure 4.97: Results of Gene $YNR016c$ obtained by the solution of original and extended model and approximately generated by Euler method with $h_k = 1/6$

4.3.2.4 Discussion

The calculated gene-expression predictions from this numerical experience using real-world data can be presented and explained with possible underlying reasons in the following way:

- It seems that the computational results have smoothed the experimental results. Due to this smoothing, the trends of the expression values becomes visible, which the models then extrapolate to the future,
- On the other hand, Euler method gives approximate results more close to experimental values in the first iterations; then, it holds that the higher the degree of the methods, the worse the approximated time series results in earlier iterations,
- The strengths of higher degree methods are in learning: the more convincing the model itself is, the better the higher-degree methods become (in terms of prediction error at various times of the iteration). Especially, if the quality of the models encounters future features (e.g., via periodicity, growth rate of the model functions, etc.), the predictive power is considered to increase.

  Here, the model class selection should at first consider the expected behavior of the modeled system. In fact, a cell cycle is evidently a cyclic process and will not have a fixed point but it is likely to converge a cyclic attractor. As the possible choices for further real-world problems, either we can select a wider model class which can model the cyclic behavior or we can select a homeostasis problem like glucose regulation mechanism. Furthermore, we can model the deviation from an expected periodic attractor.

  As another approach, the discrete algorithmical technique and analysis with polytopes, that is presented in [11, 36, 37, 39, 55], can be employed in future as a methodology to study the qualitative behavior of the dynamics.
- Actually, in the considered subnetwork, the selected data set is sparse and also very small with respect to the whole network so that one cannot conclude very much. A bigger subnetwork or the whole network can be considered for the future improvements of this real-world example,
- Accuracy is anyway just one of our goals, the other one is stability and expresses itself in some regularization and smoothing, which is especially true for possible *outliers* [157].

- We note that there can be outliers in the experimental data, and our higher degree methods serve to dampen them within our modeling and prediction. Data often contain outliers or influential observations. Outliers can be defined as one or more than one observation that are inconsistent with the rest of the data. Let us notice that, they may come from the errors of measurements and intrinsic variability. Existence of outliers in the data causes errors in parameter estimations, a misleading result and, hence, a less useful analysis. Robust methods from *statistics* can be used to deal with such kind of data together with the optimization tools as MARS and conic MARS (for details see [157]). In this way, methods from statistical learning, inverse problems and optimization are, and will further be, combined in order to obtain more efficient and stable results in the presence of outliers.

- Biologically, since we selected genes of our small network as landmarks which are the highly expressed genes in their cell-cycle phases, those genes may effect the production of some other proteins (genes) in a positive or negative way. So, continuously increasing production may happen. Also, we tried to model the whole metabolism (system) by looking at a small number of genes which are all highly expressed.

For the future improvement of the obtained results, we suggest the following methodology in order to obtain better results for this real-world application:

Since the predicted gene-expression values do not lie in a bounded region, and provided that for biological reasons unstable solutions refers to an unsatisfactory fitting, more refined statistical learning has to be performed [76]. Furthermore, additional biological knowledge has to be included by the help of experimentalist and biologists in order to specify the representative constraints and bounds for the restriction of the network and so the solution space. Moreover, it is not only a biological problem - but also a parametric one on getting an efficiency frontier or efficiency plane [139, 140]. All these kinds of analysis are included in a so-called *preprocessing* step which should be performed carefully with the data before any further task is tackled. At this step, the selection of parameters is very important and it greatly effect the solutions, especially, for the large-scaled gene-expression data, as stated with results in [158].

Because of the detected unboundedness of the generated discrete orbit, which can mean a contradiction, the studied model needs to be changed and, in fact, improved by doing necessary refinements or by considering new classes of model functions [42]. In this sense, we can

add a penalty term to the studied objective function in the sense of *Tikhonov regularization* as it is studied with the same data in [150], or we can consider a nonlinear model. In addition to these approaches, we can use *multivariate adaptive regression splines* based models as mentioned in Chapter 5 and studied with its robust conic version in [49]. They are "adaptive", i.e., they are statistically and, now, mathematically, "learning" to get into the given data set. In this thesis, we support this learning by mathematical methods, especially, by optimization.

CHAPTER 5

EXTENSIONS TOWARDS THE INCLUSION OF UNCERTAINITY THROUGH ROBUSTIFICATION

The questions about robustness come to the minds when an optimization problem is subject to uncertain data. The existence of uncertainty arise naturally in most of the real-life phenomena. Therefore, robust optimization has important applications and gains a lot of attention by scientists in recent years.

The data coming from a real-world system may include noise, e.g., both in input and output variables of the corresponding regression (fitting) problem. This means that the data of the regression problem are not exactly known or may not be exactly measured, or the exact solution of the problem may not be carried out because of intrinsic inaccuracy of the devices [166]. Also, the data may contain small changes by variations in the optimal experimental design. All these situations results uncertainty in the objective function and in possible constraints of the corresponding optimization problem. Various algorithms (see [73, 74, 79, 82, 165, 166, 168, 170] and references therein) are defined and combined from the important robust optimization method developed in [159, 161, 162, 167] in order to deal with this problem.

Robust Optimization has gained importance both theoretically and practically as a modeling framework for immunizing against parametric uncertainties in mathematical optimization. As a modeling methodology, robust optimization treats optimization problems in which data are uncertain, and are only known to belong to some uncertainty set, except for outliers. Robust optimization aims to find an optimal or near optimal solution which is feasible for every possible realization of the uncertain scenarios [165, 170]. This approach makes the optimization model robust with respect to constraint violations by solving robust counterparts

of these problems within pre-specified uncertainty sets for the uncertain parameters [114, 169]. For the worst-case realization of those uncertain parameters, the mentioned counterparts are solved based on appropriately determined uncertainty sets for the random parameters. Robust optimization problems can be solved more efficiently if the considered uncertainty set has a special shape geometrically, like polyhedral or ellipsoid (see [79] and related cited references therein). When the ellipsoidal uncertainty is considered, robustification process will produce better results, on the other hand it increases the complexity of the optimization problem [166].

5.1 Robust Optimization

Optimization gains a lot importance in the recent years in various fields like engineering, finance and control design. In most of the applications from these fields, it is assumed to have complete knowledge of the data of the optimization problem, which means that the input data are assumed to known exactly and equal to some nominal values in developing models. However, there may be significant sensitivity of the solutions to the perturbations in the parameters of the problem, thus, often a computed solution is highly infeasible or suboptimal. As a result, optimization influenced by parameter uncertainty is a central problem of the scientists in mathematical programming, and there is certainly a need to overcome uncertain data arises to develop models when optimization results are combined within real-world applications [163, 164, 165]. The scope of robust optimization is to find an optimal or near optimal solution that is feasible for every possible realization of the uncertain scenarios [165].

The general optimization problem under uncertainty is stated as follows [79]:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \boldsymbol{\alpha}^T \mathbf{x}, \\ \text{s.t.} \quad & f_i(\mathbf{x}, \mathbf{D}_i) \geq 0 \quad \forall i \in I, \\ & \mathbf{x} \in X, \end{aligned} \tag{5.1}$$

where $\mathbf{x}$ is the design vector, $\boldsymbol{\alpha}$ is a given vector of coefficients of the objective function, $f_i(\mathbf{x}, \mathbf{D}_i)$ $(i \in I)$ are given constraint functions, X is a given set and $\mathbf{D}_i$ $(i \in I)$ is the vector of random coefficient. In [160, 161, 167, 168] an important further step is taken to develop the theory for robust optimization. There, the below robust optimization problem is proposed to

be solved

$$\begin{aligned} &\max_{\mathbf{x}} \quad \boldsymbol{\alpha}^T \mathbf{x}, \\ &\text{s.t.} \quad f_i(\mathbf{x}, \mathbf{D}_i) \geq 0 \quad \forall \mathbf{D}_i \in U_i, \quad \forall i \in I, \\ &\qquad \mathbf{x} \in X, \end{aligned} \tag{5.2}$$

in which U_i ($i \in I$) denotes the given uncertainty sets corresponding to the i^{th}-constraint. Here, the aim is to find a solution of the stated problem in (5.2) which "immunizes" the problem (5.1) against parameter uncertainty. In the robust optimization literature, the uncertainty sets under consideration has the following standard types [160, 161, 167, 168], when the constraints $f_i(\mathbf{x}, \mathbf{D}_i)$ ($i \in I$) are taken linear as $\mathbf{Ax} - \mathbf{b}$ with $[\mathbf{A}, \mathbf{b}] \in U$ and $U_i = \{[\mathbf{a}_i, \mathbf{b}_i] \mid [\mathbf{A}, \mathbf{b}] \in U\}$, where

$$U_i = \{[\mathbf{a}_i, \mathbf{b}_i] = [\mathbf{a}_i^0, \mathbf{b}_i^0] + \sum_{k=1}^{K} u_k [\mathbf{a}_i^k, \mathbf{b}_i^k] \mid \mathbf{u} \in Z_i\}; \tag{5.3}$$

here, the set Z_i ($i \in I$) determines what type of uncertainty set we have. These sets may be one of the following:

$$\begin{aligned} \textit{box uncertainty set}: &\qquad Z_i = \{\mathbf{u} \in \mathbb{R}^K \mid \|\mathbf{u}\|_\infty \leq 1\}, \\ \textit{convex combination of scenarios}: &\qquad Z_i = \{\mathbf{u} \in \mathbb{R}^K \mid u_i \geq 0 \ (i = 1, 2, \ldots, K), \ \ \mathbf{e}^T \mathbf{u} \leq 1\}, \\ \textit{ellipsoid uncertainty set}: &\qquad Z_i(\mathbf{c}, \boldsymbol{\Sigma}) = \{\mathbf{u} \in \mathbb{R}^K, \ \boldsymbol{\Sigma}^{1/2}\mathbf{u} + \mathbf{c} \mid \|\mathbf{u}\|_2 \leq 1\}, \end{aligned} \tag{5.4}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}$ a symmetric nonnegative definite configuration matrix, $\mathbf{c} \in \mathbb{R}^K$ is the center of the ellipsoid and $\boldsymbol{\Sigma}^{1/2}$ is any matrix square root [171]. In (5.4), the first two uncertainty set definitions belongs to the polyhedral uncertainty type.

5.2 Robustified Process Version of Generalized Partial Linear Model Approach

In the recent paper [49], Weber, Ozmen, Cavusoglu and Defterli have presented a newly developed robust conic GPLM method with a real-world application in finance to predict the default probabilities in emerging markets. Additionally, as a further possible and new application field of robust conic GPLM, regulatory network models, e.g., eco-finance network and gene-environment network models are discussed in the introductory level. This new and challenging application field is introduced firstly in this work as the basis of this section of

the thesis and the corresponding formulations are initiated originally. The new technique of solving and optimizing the models that contain nonlinearity and uncertainty is discussed in [49] by using robust conic GPLM and robust conic MARS. An implementation on a different, especially, time-dependent area, namely, regulatory systems is introduced newly where such systems appear in environmental protection, education, system biology, medicine, financial sector, banking.

In the following part of this thesis, it is newly demonstrated [49] that the GPLM, and in fact, conic GPLM and robust conic GPLM approaches can also be implemented with the dynamical modeling of target-environment regulatory systems (see [11, 27, 29, 30, 35, 36, 39, 42, 43, 44, 46, 48, 59] and the references therein), that are considered as a subclass of regulatory networks, in order to obtain better results for the system identification. This subclass also contains eco-finance networks [30, 48] and gene-environment networks.

Regulatory networks usually contain a large number of variables and parameters, that brings complexity into the system. Therefore, in the study of such systems, there is always a need for advanced methods, which will reduce the complexity, produce more efficient and stable solutions and make it easier to deal with the problem.

Target-environment regulatory systems appear in many application areas like financial sector, economy, environmental sciences, computational biology, medicine in which they are often referred to gene-environment or eco-finance networks. One of our examples in this context is given by the process of the Kyoto Protocol (see [30] and its references). Modeling and prediction of such regulatory systems and the problem of identifying the regulating effects and interactions between the targets and other components of the network have a significant importance in the mentioned areas [11, 27, 29, 30, 35, 36, 39, 42, 43, 44, 46, 47, 48, 59].

As one of the modeling approaches applied to the target-environment regulatory networks, the time-autonomous system of ordinary differential equations has been earlier introduced and studied in [27, 35, 59]. By using the *process version* of GPLM, the same system of equations form can be reformulated in the following way [49]:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}), \tag{5.5}$$

where $F_i : \mathbb{R}^d \to \mathbb{R}$ are nonlinear coordinate functions of $\mathbf{F}$ in $X_1, X_2, \ldots, X_d$ ($d = m + n$), and $\mathbf{X} = \mathbf{X}(t) = (\boldsymbol{X}^T, \boldsymbol{T}^T)^T$ with time $t \in I$ and the interval $I = (a, b) \subseteq \mathbb{R}$. In that model, the

first n components of the d-vector $\mathbf{X} = (X_1, X_2, \ldots, X_n, T_1, T_2, \ldots, T_m)^T$ denote expression or concentration values of the n targets (also in the sense of players or genes) in the network, whereas the remaining m components denote the concentrations of environmental factors at a time t. Furthermore, $\dot{\mathbf{X}}$ stands for the change rates of $\mathbf{X}$ in time. The parameters appearing in the function are identified by using the experimental data vectors $\bar{\mathbf{X}}$, which are coming from real-world experiments and environmental measurements at the sample times [27, 35, 36, 39, 59]. A class of models has been derived from this idea in the papers [11, 29, 36, 42, 43, 46], where we can represent their generalized multiplicative form with our GPLM approach as follows [49]:

$$\dot{\mathbf{X}} = \mathbf{M}(\mathbf{X})\mathbf{X}, \quad \text{with} \quad \mathbf{X} := \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{T} \end{pmatrix}, \quad \mathbf{M}(\mathbf{X}) := \begin{pmatrix} \mathbf{M}_1(\boldsymbol{X})_{n\times n} & \mathbf{M}_1(\boldsymbol{T})_{n\times m} \\ \mathbf{M}_2(\boldsymbol{T})_{m\times n} & \mathbf{M}_2(\boldsymbol{X})_{m\times m} \end{pmatrix}. \tag{5.6}$$

While the $(n\times 1)$-vector X represents the expression levels of targets, $(m\times 1)$-vector T consists of environmental factors which affect the targets in the network. The weight functions that represent these interactions are the entries of the $(d\times d)$-matrix $\mathbf{M}(\mathbf{X})$ and they contain parameters to be estimated. In the above representation $\mathbf{M}$ is written as the chosen block structure. The matrix $\mathbf{M}(\mathbf{X})$ is called as network matrix whose entries can be polynomial, trigonometric, exponential, logarithmic, hyperbolic, spline, and it can be identified by solving the following least-squares (or maximum likelihood) estimation problem:

$$\min_{\boldsymbol{\rho}} \sum_{k=0}^{N-1} \left\| \mathbf{M}_{\boldsymbol{\rho}}(\bar{\mathbf{X}}^{(k)})\bar{\mathbf{X}}^{(k)} - \dot{\bar{\mathbf{X}}}^{(k)} \right\|_2^2, \tag{5.7}$$

N being the number of experiments and $\bar{\mathbf{X}}^{(k)}$ denotes the experimental data obtained at the k^{th}-sample time. In the above problem, $\boldsymbol{\rho}$ is some vector of unknowns, especially, parameters, involved in the functional form of $\mathbf{M} = \mathbf{M}_{\boldsymbol{\rho}}$, and $\dot{\bar{\mathbf{X}}}^{(k)}$ is some difference quotient of the values $\bar{\mathbf{X}}^{(k)}$ [36]. The dynamics of the system is described by matrices $\mathbf{M}(\mathbf{X})$ which are also the basis for testing the goodness of data fitting and prediction, and of a stability analysis [11, 27, 29, 30, 36, 39, 42, 43, 44, 46, 48].

Identification of such regulatory networks from given real-world data is an important mathematical problem to be solved both theoretically and computationally, especially, when there exist noise and uncertainty in the data [48]. In case of having a large number of variables and parameters to be identified, and nonlinear functions in the entries of the network matrix $\mathbf{M}(\mathbf{X})$, there is an increase in the complexity of such regulatory systems. Since a GPLM divides a nonlinear model into two parts and gives us the opportunity to study on the linear and

the nonlinear part separately, this idea can also be implemented for the nonlinear dynamical models of eco-finance networks as a subclass of target-environment regulatory systems. In that case, any dynamical model represented by the system of ordinary differential equations in the form of Eq. (5.6) may have linear and nonlinear entries together inside of $\mathbf{M}(\mathbf{X})$. It is assumed that nonlinear effects may come through the environmental factors. Therefore, we apply a process version of GPLM approach for the optimization of such a kind of dynamical systems, by considering all the linear entries of $\mathbf{M}(\mathbf{X})$ collected in one term and the remaining nonlinear entries in the second term, in the absence of collinearity between the independent variables $\boldsymbol{X}$ and $\boldsymbol{T}$ as mentioned in Subsubsection 2.2.1.1. In this way, we newly adopt the formulation given with Eq. (2.10) in Section 2.2 for our dynamical system defined in Eq. (5.6) which can be rewritten as

$$\begin{pmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{T}} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_1(\boldsymbol{X}) & \mathbf{M}_1(\boldsymbol{T}) \\ \mathbf{M}_2(\boldsymbol{T}) & \mathbf{M}_2(\boldsymbol{X}) \end{pmatrix} \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{T} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_1(\boldsymbol{X}) \\ \mathbf{M}_2(\boldsymbol{T}) \end{pmatrix} \boldsymbol{X} + \begin{pmatrix} \mathbf{M}_1(\boldsymbol{T}) \\ \mathbf{M}_2(\boldsymbol{X}) \end{pmatrix} \boldsymbol{T} := \begin{pmatrix} \boldsymbol{\beta}_1^T \mathbf{X} + \varsigma_1(\mathbf{T}) \\ \boldsymbol{\beta}_2^T \mathbf{X} + \varsigma_2(\mathbf{T}) \\ \vdots \\ \boldsymbol{\beta}_d^T \mathbf{X} + \varsigma_d(\mathbf{T}) \end{pmatrix}. \tag{5.8}$$

Therefore, for each row of the matrix representation in (5.8), we represent the process version of the GPLM formulation from Eqn (2.10) in Section 2.2 in the following way:

$$\dot{\mathbf{X}}_i = \boldsymbol{\beta}_i^T \mathbf{X} + \varsigma_i(\mathbf{T}) \quad (i = 1, 2, \ldots, d). \tag{5.9}$$

In the above row-wise expression, the linear terms of the each row of the system in (5.6), so in (5.8), are collected into the *linear part* $\boldsymbol{\beta}_i^T \mathbf{X}$ which depends on $\mathbf{X}$ as the vector of variables of the linear terms; similarly, the nonlinear terms are collected separately into the *nonlinear part* $\varsigma_i(\mathbf{T})$ which depends on $\mathbf{T}$ as the vector of variables of the nonlinear terms. Here, each $\boldsymbol{\beta}_i$ is a vector of parameters corresponding to the *linear part* and similarly, $\boldsymbol{\alpha}_i$ is a vector of parameters corresponding to the *nonlinear part* of the above expression. Therefore, $\boldsymbol{\rho} = (\boldsymbol{\beta}^T, \boldsymbol{\alpha}^T)^T$ can be expressed as the vector of unknown parameters appearing inside of $\mathbf{M}(\mathbf{X})$ which can be collected separately in the described way.

In this thesis study we additionally bring the following new ideas in the solution of the new modeling problem that is implemented above for regulatory networks:

For the solution process of newly implemented process version of GPLM approach used for modeling the dynamics of regulatory networks, one can use any appropriate linear program-

ming technique for the estimation of linear part of the model described in (5.9) and a nonlinear programming method for the nonlinear part correspondingly. We can offer to use, in our future applications with a given data, previously mentioned MINLP (in Section 2.3) for the nonlinear part of the estimation of GPLM together with a chosen linear programming technique according to the form of the network matrix $\mathbf{M}$.

An advanced case for target-environment networks takes place, when the entries of the matrix $\mathbf{M}$ contain spline functions, then process version of conic GPLM approach [173] can be used. The least-squares problem in (5.7) can be converted to the conic form and solved by CQP mentioned in (2.3).

Moreover, when there exist uncertainty in the expression data, where the uncertainty sets are defined like in (5.4), then process version of robust conic GPLM technique [49] can be applied and solved by appropriate linear and nonlinear programming (optimization) tools in order to study a robustification of our target-environment and gene-environment networks. Also, further comparison of the estimation results can be done by considering different methods, like GSIP extension of MIP (mentioned in (2.3)), in the solution of the same robustification problem. Thus, for each row of the matrix equation in (5.8), we represent the process version of the robust GPLM case of (5.9) in the subsequent manner [49]:

$$\dot{\breve{\mathbf{X}}}_i = \boldsymbol{\beta}_i^T \breve{\mathbf{X}} + \varsigma_i(\breve{\mathbf{T}}) \quad (i = 1, 2, \ldots, d). \tag{5.10}$$

The right-hand sides of Eqs. (5.9) and (5.10) have the descriptions of our time-continuous dynamics, which can be regarded and further analyzed as *normal forms* in the sense of, e.g., singularity theory, catastrophe theory, differential equations and optimization theory [104, 135, 137, 172]. Mathematically, normal forms can be considered for finding an additive decomposition of the right-hand side where the terms are ordered according to their degrees. Moreover, they give rise to time-discretized dynamics in very particular block structures studied in [11, 42, 46].

One can notice that, Eq. (5.10), which is different from Eq. (5.9), handles uncertainty, and the same is true for the time-discretized versions. In Eq. (5.10), uncertainty is included both in the input and output variables. In order to define and interpret derivatives of difference quotients in the presence of uncertainty, i.e., in the set-valued case, different concepts exist (see [29, 39, 43, 44, 46, 47]).

By this part of the thesis, we have introduced a new and promising implementation area of GPLM by a dynamical modeling of regulatory networks, which also include eco-finance networks and gene environment networks. By using GPLM, better results can be obtained for the parameter estimation and anticipation of these regulatory systems where they also arise in the financial sector, for example, Basel II and Basel III standards for banks are the regulating effects ("regulatory") on risk and capital management, bank capital adequacy and bank liquidity of the countries in the world ("network").

In the case of existence of uncertainty and noise in real-world data, this new model approach gains importance to reduce complexity and variance of estimation. In future studies, we will work on the newly introduced process version of GPLM for modeling, optimization and robustification of target-environment networks together with real-world applications to validate. Since there are different kinds of estimation methods for GPLM, we can choose the appropriate one according to the form of the entries of network matrix $\mathbf{M}$ [49].

CHAPTER 6

CONCLUSION

The analysis of huge amounts of time-series gene expression data, which are obtained from DNA-microarray chip experiments, and reconstruction of a genetic regulatory network from these data are a challenging problem, which has significant application areas. In this problem, one of the difficulties is that the data sets have a huge number of genes but a limited number of sampling time points. Therefore, there is an important need for effective mathematical models and efficient numerical algorithms for inferring genetic regulatory networks using such data sets [2, 3].

In this thesis, we introduce and analyze time-discrete target-environment regulatory systems, especially, for gene-environment networks in which the target variables represent the expression levels of the n genes, whereas the m environmental items stand for external factors (e.g., toxins or radiation). This thesis study widens the existing mathematical toolbox by introducing other numerical schemes of time-discretization into the study where they all belong to the higher order explicit Runge-Kutta methods. We newly derive and implement 3^{rd}-order Heun's method and 4^{th}-order classical Runge-Kutta method together with their formulations for the considered dynamic model class and also corresponding matrix algebras.

Beyond these formulations and obtained algebras of the schemes, we apply them on two different sets of gene-expression data, i.e., firstly, an artificial data set containing 4 genes with 4 sample times and secondly, a real-world data set containing 26 genes with 17 sample times as a subnetwork of a huge gene-environment network. In order to see the performance of these newly considered schemes together with the existing Euler and 2^{nd}-order Heun's method, we perform two illustrative examples with these two kinds of data sets. We investigate the behavior of all of these explicit Runge-Kutta method class with respect to different choices

of step-sizes and also to various perturbations. In this way, comparisons among these four numerical methods are studied in many ways and detailed discussions about the obtained results are provided in Subsubsections 4.3.1.3 - 4.3.1.5 and also in Subsubsection 4.3.2.4.

In our numerical examples studied in Chapter 4, we consider the linear dynamical model $\dot{\mathbf{E}} = \mathbf{ME}$ (to express the gene interactions) for both of our illustrative examples and then formulate corresponding MINLP problems as our optimization problems for the identification of parameters appearing in that models. Therefore, we present and study the MINLP problems for two cases, e.g., by including negative regulation effect in the network or by omitting this effect. We recall these two types of MINLP problems as *original* MINLP model and *extended* MINLP model. In the constructions of these problems the choices of the bounds in the constraints play an important role in the solution. In our application with artificial data these bound and constraints are already specified in [36, 37]. But, in our second application with real-world data, we tried to identify and select the corresponding bounds of the constraints according to the biological behavior of the considered data and so the network. For this purpose, we first selected a subnetwork from the huge network of this real-world data, representatively, for diminishing the computational complexity. Then, we worked on some statistical and biological properties of the genes in this selected subnetwork and calculate their indegree, outdegree values and degradation rates to be used as the bounds in the corresponding MINLP problems. After solving both original and extended optimization problems, we apply our class of numerical schemes to generate the approximate time series of gene-expression results for the further time levels and try to predict the long-time behavior.

From the results that we obtained from two examples, we can conclude that the performance of newly studied higher order numerical methods are better than the Euler and 2^{nd}-order Heun's method in terms of accuracy and smoothing, if the chosen model is stable, compatible with the general behavior and the considered data is not noisy or without outliers [157].

For our first example with artificial data, reducing the step-size have improved the rate of convergence in Euler and 3^{rd}-order Heun's method but did not effect much the results of 4^{th}-order classical Runge-Kutta method since it has already smoothen the results of all genes and let them to reach the limit point. For the 2^{nd}-order Heun's method, it improves the results of the genes which have alternating behavior and make them to converge to the limit point. Our perturbation analysis in Subsection 4.3.1 gives us a possible interval for an allowed perturbation

value which will not change the general behavior of the gene inside the considered numerical scheme and preserving the smooth and converging behavior.

The considered higher-order Runge-Kutta methods takes the values at each discrete time, from other neighbouring discrete times (both by past and future time points), which has a smoothing effect. These higher-order methods contribute to the entire basic concept of the thesis in the regard of regularization, rarefication and smoothing and they are borrowing from the theory of inverse problems, which is now understood and applied in the sense of dynamics.

In this thesis work, we finally introduced into the involvement of uncertainty in gene-expression data where the uncertainty or errors are coming from microarray experiments and measurements of the environment. Robustification is announced in order to deal with the inclusion of uncertainty in the model. Here, we even open a long-term research project by introducing a new and challenging implementation of a generalized partial linear modeling approach on gene-environment networks and its robustified version.

By this extended numerical methods for the time discrete dynamics of regulatory systems and introduced new ideas of modeling with GPLM approach, including its robustified version in the case uncertainty, we aim at being better prepared for the modeling, prediction, stability, regularization and robustification of our networks for a better service in the real-world areas. These areas are, e.g., the study of gene and metabolic networks, diminishing negative effects of the environment (also life style and living conditions) on health, surveying the effects of medical treatments, Kyoto protocol and other environmental campaigns and even financial markets.

Future study;

In our future work which we plan and propose, we continue to investigate our real-world example with an appropriate regularization method in order to overcome unbounded gene-expression results and to represent the behavior of the considered real data in a much better way as explained in Subsubsection 4.3.2.4. Moreover, we will include environmental effects into our applications by introducing a nonlinear dynamical model of the type $\dot{\mathbf{E}} = \mathbf{M}(\mathbf{E})\mathbf{E}$. By the help of a detailed *learning process* combined with a stable model, our new numerical schemes can produce better predictions for the long-term behavior these systems (networks).

The stability research of the models could be continued by us even more systematically, which

is left as a possible future research. Indeed, in the earlier studies on stability [1, 11, 36, 37, 39, 55], both stable and unstable behaviors are investigated based on parametric variations of the model of the dynamics and employing an Euler discretization. In those studies, analytically, Lyapunov functions and their discrete orbits are followed which are generated by stepwise applying system matrices (in the sense of after discretization) on compact neighborhoods of the zero state vector. The authors also investigated the orbits boundedness (stability) or unboundedness (instability). Since the (numerically) determined boundary between stability and instability can be detected by using any desired precision [39], this approach can be undertaken in future with our newly elaborated 3^{rd}-order Heun's discretization scheme. In this respect, some further studies can be carried out by extending the work performed in this thesis already.

As another aspect of future study, we will obtain the detailed formulations of the (process version of) GPLM approach and its robustification that are newly introduced in Chapter 5. Then, we will apply them on a given set of data to validate the performance and efficiency together with the new directions and ideas presented in the last part of the same chapter. Herewith, we naturally come into our domain of dynamical modeling and overcoming of uncertainty as introduced in Chapter 5.

REFERENCES

[1] Radde, N. *Modeling non-linear dynamic phenomena in biochemical networks*, Ph.D. Thesis, Faculty of Mathematics and Natural Sciences, University of Köln, (2007).

[2] Jong, H.D. *Modeling and simulation of genetic regulatory systems: A literature review*, Journal of Computational Biology, **9(1)**, 67-103, (2002).

[3] Bansal, M., Belcastro, V., Ambesi-Impiombato, A. and di Bernardo, D. *How to infer gene networks from expression profiles*, Molecular Systems Biology, **3**, 78, (2007).

[4] Bolouri, H. and Davidson, E.H. *Modeling transcriptional regulatory networks*, BioEssays, **24**, 1118-1129, (2002).

[5] Hasty, J., McMillen, D., Isaacs, F. and Collins, J.J. *Computational studies of gene regulatory networks: in numero molecular biology*, Nature Reviews Genetics, **2**, 268-279, (2001).

[6] Huang, S. *Gene expression profiling, genetic networks and cellular states: an integrating concept for tumorigenesis and drug discovery*, Journal of Molecular Medicine, **77**, 469-480, (1999).

[7] Smolen, P., Baxter, D.A. and Byrne, J.H. *Modeling transcriptional control in gene networks - methods, recent results, and future directions*, Bulletin of Mathematical Biology, **62**, 247-292, (2000).

[8] Werhli, A., Grzegorczyk, M. and Husmeier, D. *Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks*, Bioinformatics, **22(20)**, 2523-2531, (2006).

[9] Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. *Network Flow: Theory, Algorithms and Applications*, Prentice Hall, New Jersey, (1993).

[10] Gebert, J., Lätsch, M., Ming Poh Quek, E. and Weber, G.-W. *Analyzing and optimizing genetic network structure via path-finding*, Journal of Computational Technologies, **9(3)**, 3-12, (2004).

[11] Taştan, M. *Analysis and prediction of gene expression patterns by dynamical systems, and by a combinatorial algorithm*, M.Sc. Thesis, Institute of Applied Mathematics, Middle East Technical University, Ankara, (2005).

[12] Kauffman, S. *Metabolic stability and epigenesis in randomly constructed genetic nets*, Journal of Theoretical Biology, **22**, 437-467, (1969).

[13] Albert, R. and Othmer, H.G. *The topology of the regulatory interactions predict the expression pattern of the segment polarity genes in Drosophila melanogaster*, Journal of Theoretical Biology, **223**, 1-18, (2003).

[14] Bornholdt, S. *Less is more in modeling large genetic networks*, Science, **310(5747)**, 449-451, (2005).

[15] Li, F., Long, T., Lu, Y., Ouyangm, Q. and Tang, C. *The yeast cell-cycle network is robustly designed*, Proceedings of the National Academy of Sciences, **101**, 4781-4786, (2004).

[16] Liang, S., Fuhrman, S. and Somogyi, R. *Reveal, a general reverse engineering algorithm for inference of genetic network architectures*, Pacific Symposium in Biocomputing, **3**, 18-29, (1998).

[17] Shmulevich, I., Saarinen, A., Yli-Harja, O. and Astola, J. *Inference Of Genetic Regulatory Networks Via Best-Fit Extension*, Springer, US, (2003).

[18] Thieffry, D. and Thomas, R. *Qualitative analysis of gene networks*, Pacific Symposium in Biocomputing, **3**, 77-88, (1998).

[19] Thomas, R. and D'Ari, R. *Biological Feedback*, CRC Press, Boca Raton, FL, USA, (1990).

[20] Thomas, R., Thieffry, D. and Kauffman, M. *Dynamical behaviour of biological regulatory networks - I. biological role of feedback loops and practical use of the concept of the loop-characteristic state*, Bulletin of Mathematical Biology, **57**, 247-276, (1995).

[21] Shmulevich, I., Dougherty, E.R. and Zhang, W. *From Boolean to probabilistic Boolean networks as models of genetic regulatory networks*, Proceedings of the IEEE, **90(11)**, 1778-1792, (2002).

[22] Murphy, K. and Mian, S. *Modelling gene expression data using dynamic Bayesian networks*, Technical report, Computer Science Division, University of California, Berkeley, CA, USA, (1999).

[23] Friedman, N., Linial, M., Nachman, I. and Pe'er, D. *Using Bayesian networks to analyze expression data*, Journal of Computational Biology, **7(3-4)**, 601-620, (2000).

[24] Husmeier, D. *Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks*, Bioinformatics, **19(17)**, 2271-2282, (2003).

[25] Yılmaz, F.B. *A mathematical modeling and approximation of gene expression patterns by linear and quadratic regulatory relations and analysis of gene networks*, M.Sc. Thesis, Institute of Applied Mathematics, Middle East Technical University, Ankara, (2004).

[26] Glass, L. *Classification of biological networks by their qualitative dynamics*, Journal of Theoretical Biology, **54**, 85-107, (1975).

[27] Chen, T., He, H.L. and Church, G.M. *Modeling gene expression with differential equations*, Proceedings of Pacific Symposium in Biocomputing, **4**, 29-40, (1999).

[28] Gebert, J., Radde, N. and Weber, G.-W. *Modelling gene regulatory networks with piecewise linear differential equations*, European Journal of Operational Research, **181(3)**, 1148-1165, (2007).

[29] Weber, G.W., Uğur, Ö., Taylan, P. and Tezel, A. *On optimization, dynamics and uncertainty: a tutorial for gene-environment networks*, Discrete Applied Mathematics, **157(10)**, 2494-2513, (2009).

[30] Weber, G.W., Kropat, E., Tezel, A. and Belen, S. *Optimization applied on regulatory and eco-finance networks-survey and new developments*, Pacific Journal of Optimization, **6(2)**, 319-340, (2010).

[31] Weber, G.-W., Kropat, E., Akteke-Öztürk, B. and Görgülü", Z.K. *A Survey on OR and Mathematical Methods Applied on Gene-Environment Networks*, Central European Journal of Operations Research, **17(3)**, 315-341, (2009).

[32] Weber, G.-W., Alparslan Gök, S.Z. and Dikmen, N. *Environmental and life sciences: Gene-environment networks-optimization, games and control-a survey on recent achievements*, Journal of Organizational Transformation and Social Change, **5(3)**, 197-233, (2008).

[33] Öktem, H. *A survey on piecewise-linear models of regulatory dynamical systems*, Nonlinear Analysis, **63**, 336-349, (2005).

[34] Kaderali, L. and Radde, N. *Inferring gene regulatory networks from expression data*, volume 1 of Studies in Computational Intelligence, chapter 2, Springer-Verlag, Berlin, (2007).

[35] Sakamoto, E. and Iba, H. *Inferring a system of differential equations for a gene regulatory network by using genetic programming*, In: Proceedings of Congress on Evolutionary Computation, 720-726, (2001).

[36] Gebert, J., Lätsch, M., Pickl, S.W., Weber, G.-W. and Wünschiers, R. *Genetic networks and anticipation of gene expression patterns*, In: Computing Anticipatory Systems: CASYS(92)03 - Sixth International Conference, AIP Conference Proceedings, vol.**718**, 474-485, (2004).

[37] Gebert, J., Lätsch, M., Pickl, S.W., Weber, G.-W. and Wünschiers, R. *An algorithm to analyze stability of gene-expression pattern*, Discrete Applied Mathematics, **154(7)**, 1140-1156, (2006).

[38] Taştan, M., Pickl, S.W. and Weber, G.-W. *Mathematical modeling and stability analysis of gene-expression patterns in an extended space and with Runge-Kutta discretization*, In: Proceedings of Operations Research (Bremen, September 2005), 443-450, (2006).

[39] Uğur, O., Pickl, S.W., Weber, G.-W. and Wünschiers, R. *An algorithmic approach to analyze genetic networks and biological energy production: an introduction and contribution where OR meets biology*. Optimization, **58(1)**, 1-22, (2009).

[40] Weber, G.-W. and Tezel, A. *On generalized semi-infinite optimization of genetic networks*, TOP, **15(1)**, 65-77, (2007).

[41] Yılmaz, F.B., Öktem, H. and Weber, G.-W. *Mathematical modeling and approximation of gene expression patterns and gene networks*, In: Operations Research Proceedings, F. Fleuren, D. den Hertog, P. Kort (Eds.), 280-287, (2005).

[42] Weber, G.W., Tezel, A., Taylan, P., Soyler, A. and Çetin, M. *Mathematical contributions to dynamics and optimization of gene-environment networks*, Optimization, **57(2)**, 353-377, (2008).

[43] Weber, G.-W., Taylan, P., Alparslan Gök, S.Z., Özöğür, S. and Akteke Öztürk, B. *Optimization of gene-environment networks in the presence of errors and uncertainty with Chebychev approximation*, TOP, **16(2)**, 284-318, (2008).

[44] Kropat, E., Weber, G.-W. and Pedamallu, C.S. *Regulatory networks under ellipsoidal uncertainty-optimization theory and dynamical systems*, Preprint **22**, Institute of Applied Mathematics, METU, (2009) (submitted to SIAM Journal on Optimization).

[45] Weber, G.-W., Defterli, O., Kropat, E. and Alparslan-Gök, S.Z. *Modeling, inference and optimization of regulatory networks based on time series data*, European Journal Of Operational Research, **211(1)**, 1-14, (2011).

[46] Uğur, Ö. and Weber, G.-W. *Optimization and dynamics of gene-environment networks with intervals*, Journal of Industrial Management and Optimization, **3(2)**, 357-379, (2007).

[47] Kropat, E., Weber, G.-W. and Belen, S. *Dynamical gene-environment networks under ellipsoidal uncertainty – set-theoretic regression analysis based on ellipsoidal OR*, In: Dynamics, Games and Science I, Springer Proceeding in Mathematics, (Proceedings of the conference DYNA2008, Braga, Portugal), M. Peixoto, D. Rand and A. Pinto (Eds.), Springer-Verlag 2010, ISBN: 978-3-642-11455-7, Ch. **35**, 545-571, (2010).

[48] Kropat, E. and Weber, G.-W. *Robust regression analysis for gene-environment and eco-finance networks under polyhedral and ellipsoidal uncertainty*, Preprint **2**, Institute of Applied Mathematics, Middle East Technical University, (2010) (submitted to Optimization Methods and Software).

[49] Ozmen, A., Weber, G.-W., Cavusoglu, Z. and Defterli, O. *The new robust conic GPLM Method - with an application to finance and regulatory systems: prediction of credit default and a process version*, Journal of Global Optimization, (submitted), (2011).

[50] Aster, R.C., Borchers, B. and Thurber, C.H. *Parameter Estimation and Inverse Problems*, Academic Press, New York, (2004).

[51] Hadamard, J. *Lectures on Cauchy's Problem in Linear Partial Differential Equations*, Yale University Press, New Haven, (1923).

[52] Vierstraete, A. `http://users.ugent.be/avierstr/`, last visited: July 2011.

[53] Schena, M. *DNA Microarrays*, Oxford University Press, (2000).

[54] Schena, M., Shalon, D., Davis, R.W. and Brown, P.O. *Quantitative monitoring of gene expression patterns with a complementary DNA microarray*, Science, **270**, 467-470, (1995).

[55] Gebert, J., Pickl, S., Shokina, N., Weber, G.-W. and Wünschiers, R. *Algorithmic analysis of gene expression data with polyhedral structures*, In: Proceedings of Similarity Methods (5^{th} International Workshop), B. Kröplin, S. Rudolph, J. Häcker (Eds.), ISBN 3-930683-47-4, 79-87, (2001).

[56] Ideker, T.E., Thorsson, V. and Karp, R.M. *Discovery of regulatory interaction through perturbation: inference and experimental design*, Pacific Symposium on Biocomputing, **5**, 302-313, (2000).

[57] Wagner, A. *Estimating coarse gene network structure from large-scale gene perturbation data*, Genome Research, **12**, 309-315, (2002).

[58] de Hoon, M., Imoto, S. and Miyano, S. *Inferring gene regulatory networks from time-ordered gene expression data using differential equations*, Discovery Science, 267-274, (2002).

[59] Hoon, M.D., Imoto, S., Kobayashi, K., Ogasawara, N. and Miyano, S. *Inferring gene regulatory networks from time-ordered gene expression data of Bacillus Subtilis using differential equations*, Proceedings of Pacific Symposium in Biocomputing, **8**, 17-28, (2003).

[60] van Someren, E.P., Wessels, L.F.A. and Reinders, M.J.T. *Linear modeling of genetic networks from experimental data*, In: Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (La Jolla, CA), AAAI Press, Menlo Park, CA, 355-366, (2000).

[61] Purutcuoglu, V. and Wit, E. *Bayesian inference of the complex MAPK pathway under structural dependency*, Journal of Statistical Research, **6(1)**, 1-17, (2009).

[62] Weaver, D.C., Workman, C.T. and Stormo, G.D. *Modeling regulatory networks with weight matrices*, Pacific Symposium on Biocomputing, **4**, 112-123, (1999).

[63] Friedman, N., Linial, M., Nachman, I. and Pe'er, D. *Using Bayesian networks to analyze expression data*, Journal of Computational Biology, **7**, 601-620, (2000).

[64] MedicineNet, authored: Webster's New World Medical Dictionary, `http://www.medterms.com/script/main/art.asp?articlekey=21819`, last visited: July 2011.

[65] Weber, G.-W., Alparslan-Gök, S.Z. and Söyler, B. *A new mathematical approach in environmental and life sciences: gene-environment networks and their dynamics*, Environmental Modeling & Assessment, **14(2)**, 267-288, (2009).

[66] Liu, Q., Yang, J., Chen, Z., Yang, M.Q., Sung, A.H. and Huang, X. *Supervised learning-based tagSNP selection for genome-wide disease classifications*, BMC Genomics, **9**, 1, (2007).

[67] Borenstein, E. and Feldman, M.W. *Topological signatures of species interactions in metabolic networks*, Journal of Computational Biology, **16(2)**, 191-200, (2009).

[68] Partner, M., Kashtan, N. and Alon, U. *Environmental variability and modularity of bacterial metabolic networks*, BMC Evolutionary Biology, **7**, 169, (2007).

[69] Harris, J.R., Nystad, W. and Magnus, P. *Using genes and environments to define asthma and related phenotypes: applications to multivariate data*, Clinical and Experimental Allergy, **28(1)**, 43-45, (1998).

[70] Gökmen, A., Kayaligil, S., Weber, G.W., Gökmen, I., Ecevit, M., Sürmeli, A., Bali, T., Ecevit, Y., Gökmen, H. and DeTombe, D.J. *Balaban Valley Project: Improving the quality of life in rural area in Turkey*, International Scientific Journal of Methods and Models of Complexity, **7(1)**, `www.fss.uu.nl/ms/cvd/isj/index7-1.htm`, (2004).

[71] Fox, J. *Nonparametric Regression*, in Encyclopedia of Statistics in the Behavioral Sciences, B. Everitt and D. Howell (Eds.), Wiley, London, (2005).

[72] Montgomery, D.C. and Runger, G.C., *Applied Statistics and Probability for Engineers*, John Wiley and Sons, New York, (2007).

[73] Weber, G.-W., Batmaz, I. and Özmen, A. *Robust conic quadratic programming - A robustification of CMARS*, Preprint **4**, Institute of Applied Mathematics, Middle East Technical University, (2010).

[74] Taylan, P., Weber, G.-W., Lian, L. and Yerlikaya-Özkurt, F. *On foundations of parameter estimation for generalized partial linear models with B-splines and continuous optimization*, Computers and Mathematics with Applications, **60(1)**, 134-143, (2010).

[75] Myers, R.H. and Montgomery, D.C. *Response surface methodology: Process and Product Optimization Using Designed Experiments*, Wiley Series in Probability and Statistics, Second edition, John Wiley & Sons Inc., New York, (2002).

[76] Hastie, T.J., Tibshirani, R.J. and Friedman, J. *The Elements of Statistical Learning, Data Mining, Inference and Prediction*, Springer Verlag, New York, (2001).

[77] McCullagh, P. and Nelder, J.A. *Generalized Linear Models*, Chapman and Hall, London, (1989).

[78] Müller, M. *Estimation and Testing in Generalized Partial Linear Models - A Comparive Study*, Statistics and Computing, **11**, 299-309, (2001).

[79] Özmen, A. *Robust conic quadratic programming applied to quality improvement - A robustification of CMARS*, M.Sc. Thesis, Institute of Applied Mathematics, Middle East Technical University, Ankara, (2010).

[80] Kayhan, B. *Parameter estimation in generalized partial linear models with Tikhonov regularization method*, M.Sc. Thesis, Institute of Applied Mathematics, Middle East Technical University, Ankara, (2010).

[81] Çelik, G. *Parameter estimation in generalized partial linear models with conic quadratic programming*, M.Sc. Thesis, Institute of Applied Mathematics, Middle East Technical University, Ankara, (2010).

[82] Yerlikaya, F. *A new contribution to nonlinear robust regression and classification with MARS and its application to data mining for quality control in manufacturing*, M.Sc. Thesis at the Institute of Applied Mathematics, Middle East Technical University, Ankara, (2008).

[83] Bates, D.M. and Watts, D.G. *Nonlinear Regression Analysis and Its Applications*, Wiley Series in Probability and Statistics, John Wiley and Sons Inc., Hoboken, NJ, (2008).

[84] Defterli, O., Fügenschuh, A. and Weber, G.-W. *Modern tools for the time-discrete dynamics and optimization of gene-environment networks*, Communications in Nonlinear Science and Numerical Simulations, **16(12)**, 4768 - 4779, (2011).

[85] Defterli, O., Fügenschuh, A. and Weber, G.-W. *New discretization and optimization techniques with results in the dynamics of gene-environment networks*, In: Proceedings of the 3^{rd} Global Conference on Power Control&Optimization (February 2-4, 2010, Gold Coast, Queensland, Australia), N. Barsoum, P.Vasant, R. Habash (Eds.), CD-ISBN: 978-983-44483-1-8, (2010).

[86] Ben-Tal, A. and Nemirovski, A. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPR-SIAM Series on Optimization, SIAM, Philadelphia, (2001).

[87] Hettich, R. and Kortanek, O. *Semi-infinite programming: theory, methods and applications*, SIAM Review, **35(3)**, 380-429, (1993).

[88] Hettich, R. and Zencke, P. *Numerische Methoden der Approximation und semi-infiniten Optimierung*, Tuebner, Stuttgart, (1982).

[89] Hettich, R. and Jongen, H.Th. *Semi-infinite programming: conditions of optimality and applications*, In: Optimization Techniques 2, Lecture notes in Control and Information Sci., J. Stoer (Eds.), Springer, Berlin, 1-11,(1978).

[90] Goberna, M.A. and Lopez, M.A. *Linear Semi-Infinite Optimization*, John Wiley and Sons Ltd, (1998).

[91] Reemtsen, R. and Görner, S. *Numerical methods for semi-infinite programming: a survey*, In: Semi-Infinite Programming, R. Reemtsen, J.-J. Rückmann (Eds.), 195-275, Kluwer, Boston, (1998).

[92] Still, G. *Generalized semi-infinite programming: theory and methods*, European Journal of Operational Research, **119**, 301-313, (1999).

[93] Still, G. *Semi-infinite programming: An introduction, preliminary version*, University of Twente Department of Applied Mathematics, The Netherlands, (2004).

[94] Weber, G.-W. *Charakterisierung struktureller stabilit at in der nichtlinearen optimierung*, In: Aachener Beitraege zur Mathematik 5, H.H. Bock, H.Th. Jongen and W. Plesken (Eds.), Augustinus publishing house (now: Mainz publishing house), Aachen, (1992).

[95] Weber, G.-W. *Minimization of a max-type function: Characterization of structural stability*, In: Parametric Optimization and Related Topics III, J. Guddat, H.Th. Jongen, B. Kummer and F. Nozicka (Eds.), 519-538, Peter Lang Publishing House, Bern, (1993).

[96] Weber, G.-W., Taylan, P., Ozogur, S. and Akteke- Ozturk, B. *Statistical learning and optimization methods in data mining*, In: Recent Advances in Statistics, H.O. Ayhan and I. Batmaz (Eds.), Turkish Statistical Institute Press, 181-195, (2007).

[97] Weber, G.-W. Generalized Semi-Infinite Optimization and Related Topics, Research and Exposition in Mathematics, Vol. **29**, Heldermann Verlag, Germany, (2003).

[98] Wetterling, W.W.E. *Definitheitsbedingungen für relative extrema bei optimierungsund approximationsaufgaben*, Numerische Mathematik, **15**, 122-136, (1970).

[99] Polak, E. *On the mathematical foundation of nondifferentiable optimization in engineering design*, SIAM Review, **29**, 21-89, (1997).

[100] Hettich, R. and Still, G. *Second order optimality conditions for generalized semi-infinite programming problems*, Optimization, **34**, 195-211, (1995).

[101] Klatte, D. *Stability of stationary solutions in semi-infinite optimization via the reduction approach*, In: Advances in Optimization, Lecture Notes in Economics and Mathematical Systems, W. Oettli, D. Pallaschke (Eds.), Springer, Berlin, Vol. **382**, 155-170, (1992).

[102] Guerra Vazquez, F., Rückmann, J.-J., Stein, O. and Still, G. *Generalized semi-infinite programming: A tutorial*, Journal of Computational and Applied Mathematics, **217**, 394-419, (2008).

[103] Stein, O. and Tezel, A. *The semismooth approach for semi-infinite programming under the reduction ansatz*, Journal of Global Optimization, **41**, 245-266 (2008).

[104] Jongen, H.Th., Jonker, P. and Twilt, F. *Nonlinear Optimization in Finite Dimensions-Morse Theory, Chebyshev Approximation, Transversality, Flows, Parametric Aspects*, Nonconvex Optimization and its Applications, Vol. **47**, Kluwer Academic Publishers, Boston, (2000).

[105] Özöğür-Akyüz, S. *Mathematical contribution of statistical learning and continuous optimization using infinite and semi-infinite programming to computational statistics*, Ph.D. Thesis in Institute of Applied Mathematics, Middle East Technical University, Ankara, (2009).

[106] Tezel Özturan, A. *A semismooth newton method for generalized semi-infinite programming problems*, Ph.D. Thesis in Graduate School of Natural and Applied Sciences, Department of Mathematics, Middle East Technical University, Ankara, (2010).

[107] Aardal, K., Weismantel, R. and Wolsey, L.A. *Non-standard approaches to integer programming*, Discrete Applied Mathematics, **123/124**, 5-74, (2002).

[108] Johnson, E.L., Nemhauser, G.L. and Savelsbergh, M.W.P. *Progress in linear programming-based algorithms for integer programming: An exposition*, INFORMS Journal on Computing, **12(1)**, (2000).

[109] Marchand, H., Martin, A., Weismantel, R. and Wolsey, L.A. *Cutting planes in integer and mixed integer programming*, Discrete Applied Mathematics, **123/124**, 391-440, (2002).

[110] Fügenschuh, A. and Martin, A. *Computational integer programming and cutting planes*, In: Handbooks in Operations Research and Management Science, Handbook on Discrete Optimization, K. Aardal, G. Nemhauser, R. Weismantel (Eds.), Elsevier, Amsterdam, Vol. **12**, 69-122, (2005).

[111] Garey, M.R. and Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, (1979).

[112] Faigle, U., Kern, W. and Still, G. *Algorithmic Principles of Mathematical Programming*, Kluwer, Dordrecht, (2002).

[113] Vandenberghe, L. and Boyd, S. *Connections between semi-infinite and semi-definite programming*, In: Semi-Infinite Programming, R. Reemtsen, J.J. Rückmann (Eds.), Kluwer, Boston, 277-294, (1998).

[114] Boyd, S. and Vandenberghe, L. *Convex Optimization*, Cambridge University Press, Cambridge, UK, (2004).

[115] Smith, E.M.B. and Pantelides, C.C. *A symbolic reformulation/spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs*, Computers & Chemical Engineering, **23**, 457-478, (1999).

[116] Tawarmalani, M. and Sahinidis, N.V. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software and Applications*, Kluwer Academic Publishers, Boston MA, (2002).

[117] Tawarmalani, M. and Sahinidis, N.V. *Global optimization of mixed-integer nonlinear programs: A theoretical and computational study*, Mathematical Programming, **99(3)**, 563-591, (2004).

[118] Perko, L. *Differential Equations and Dynamical Systems*, Texts in Applied Mathematics, Springer Series, New York, (1991).

[119] Weber, G.-W., Taylan, P., Akteke-Öztürk, B. and Uğur, Ö. *Mathematical and data mining contributions to dynamics and optimization of gene-environment networks*, Electronic Journal of Theoretical Physics, **4, 16(II)**, 115-146, (2007).

[120] D'Haeseler, P., Wen, X., Fuhrman, S. and Somogyi, R. *Linear modeling of mRNA expression levels during cns development and injury*, Pacific Symposium in Biocomputing, 41-52, (1999).

[121] van Someren, E.P., Wessels, L.F.A. and Reinders, M.J.T. *Linear modeling of genetic networks from experimental data*, In: Proceedings of the 8^{th} International Conference on Intelligent Systems for Molecular Biology (La Jolla, California, USA), 355-366, (2000).

[122] Wu, F.X., Zhang, W.J. and Kusalik, A.J. *Modeling gene expression from microarray expression data with state-space equations*, Pacific Symposium in Biocomputing, **9**, 581-592, (2004).

[123] Cokus, S.J., Haynor, D., Gronbech-Jensen, N. and Pellegrini, M. *Modelling the network of cell cycle transcription factors in the yeast Saccharomyces cerevisiae*. BMC Bioinformatics, **7(381)**, (2006).

[124] Kloster, M., Tang, C. and Wingreen, N.S. *Finding regulatory modules through largescale gene-expression data analysis*, Bioinformatics, **21(7)**, 1172-1179, (2005).

[125] Guthke, R., Möller, U., Hoffmann, M., Thies, F. and Töpfer, S. *Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection*, Bioinformatics, **21(8)**, 1626-1634, (2005).

[126] Yu, T. and Li, K.-C. *Inference of transcriptional regulatory network by two-stage constrained space factor analysis*, Bioinformatics, **21(21)**, 4033-4038, (2005).

[127] Vallabhajosyula, R.R., Chickarmane, V. and Sauro, H.M. *Conservation analysis of large biochemical networks*, Bioinformatics, **22(3)**, 346-353, (2006).

[128] Gustafsson, M., Hörnquist, M. and Lombardi, A. *Constructing and analyzing a largescale gene-to-gene regulatory network - Lasso constrained inference and biological validation*, IEEE Transcations on Computational Biology and Bioinformatics, **2(3)**, 254-261, (2005).

[129] Sanguinetti, G., Lawrence, N.D. and Rattray, M. *Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities*, Bioinformatics, **22(22)**, 2775-2781, (2006).

[130] Taştan, M., Ergenç, T., Pickl, S.W. and Weber, G.W. *Stability analysis of gene expression patterns by dynamical systems and a combinatorial algorithm*, Proceeedings of International Symposium on Health Informatics and Bioinformatics, 67-75, (2005).

[131] Akhmet, M.U., Gebert, J., Öktem, H., Pickl, S.W. and Weber, G.W. *An improved algorithm for analytical modelling and anticipation of gene expression patterns*, Journal of Computational Technologies, **10(4)**, 3-20, (2005).

[132] Isaacson, E. and Keller, H.B. *Analysis of Numerical Methods*, John Wiley and Sons, New York, (1966).

[133] Dubois, D.M. and Kalisz, E. *Precision and stability of Euler, Runge-Kutta and incursive algorithm for the harmonic oscillator*, International Journal of Computing Anticipatory Systems, **14**, 21-36, (2004).

[134] Ergenç, T. and Weber, G.W. *Modeling and prediction of gene-expression patterns reconsidered with Runge-Kutta discretization*, Journal of Computational Technologies **9(6)**, 40-48, (2004).

[135] Amann, H. *Differentialgleichungen*, Walter de Gruyter, Berlin, New York, (1983).

[136] Brayton, R.K. and Tong, C.H. *Stability of dynamical systems: A constructive approach*, IEEE Transactions on Circuits and Systems, **26(4)**, 224-234, (1979).

[137] Jongen, H.T. and Weber, G.-W., *On parametric nonlinear programming*, Annals of Operations Research, **27**, 253-284, (1990).

[138] Gebert, J., Öktem, H., Pickl, S.W., Radde, N., Weber, G.-W. and Yılmaz, F.B. *Inference of gene expression patterns by using a hybrid system formulation - an algorithmic approach to local state transition matrices*, In: Anticipative and Predictive Models in Systems Science I, G.E. Lasker and D.M. Dubois (Eds.), IIAS (International Institute for Advanced Studies) in Windsor, Ontario, 63-66, (2004).

[139] Özmen, A., Weber, G-W., Batmaz, I. *The new robust CMARS (RCMARS) method*, In: ISI Proceedings of 24th MEC-EurOPT 2010-Continuous Optimization and Information-Based Technologies in the Financial Sector (Izmir, Turkey), ISBN: 978-9955-28-598-4, 362-368, (2010).

[140] Taylan, P., Weber, G.-W., Yerlikaya-Özkurt, F. *Continuous optimization applied in MARS for modern applications in finance, science and technology*, ISI Proceedings of 20^{th} Mini-EURO Conference (Neringa, Lithuania), 317-322, (2008).

[141] Heath, M. *Scientific Computing: An Introductory Survey*, McGraw-Hill, (2002).

[142] Koch, T. *Rapid mathematical programming*, Ph.D. Thesis, Technische Universität Berlin, Technical Report ZIB-TR 04-58, (2004).

[143] Achterberg, T. *Constraint integer programming*, Ph.D. Thesis, Technische Universitat Berlin, (2007).

[144] Berthold, T., Heinz, S. and Vigerske, S. *Extending a CIP framework to solve MIQCPs*, Konrad-Zuse-Zentrum fur Informationstechnik, Berlin, Technical Report ZIB-TR 09-23, (2009).

[145] R. Wunderling, *Paralleler und objektorientierter simplex-algorithmus*, Ph.D. Thesis, Technische Universitat Berlin, Technical Report ZIB-TR 96-09, (1996).

[146] Wilkins, M.B. and Thomas, S.L. *The damping and reinitiation of the circadian rhythm of CO_2 output in Bryophyllum leaves in relation to their malate content*, Journal of Experimental Botany, **44(262)**, 901-906, (1993).

[147] Stein, O. *Bifurcations of hyperbolic fixed points for explicit Runge-Kutta methods*, IMA Journal of Numerical Analysis, **17**, 151-175, (1997).

[148] Mordukhovich, B.S. *Variational Analysis and Generalized Differentiation II: Applications*, Grundlehren Series (Fundamental Principles of Mathematical Sciences), Springer, (2006).

[149] Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D., Lockhart, D.J. and Davis, R.W. *A genome-wide transcriptional analysis of the mitotic cell cycle*, Molecular Cell, **2**, 65-73, (1998).

[150] Zhang, S.Q., Ching, W.K., Tsing, N.K., Leung, H.Y. and Guo, D. *A new multiple regression approach for the construction of genetic regulatory networks*, Artificial Intelligence in Medicine, **48**, 153-160, (2010).

[151] Yeung K.Y. and Ruzzo, W.L. *Principal component analysis for clustering gene expression data*, Bioinformatics, **17**, 763-774, (2001).

[152] Vellguth, M. and Wünschiers, R. *A Web-Based Application for Visualizing Gene Clusters on Metabolic Pathway Maps*, arXiv:0706.3477v1 [q-bio.GN], (2007).

[153] OrfMapper software, `http://www.orfmapper.com/`, last visited: June 2011.

[154] BioLayout Express3D, `http://www.biolayout.org/`, last visited: June 2011.

[155] IBM ILOG CPLEX Optimizer software, `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`, last visited: July 2011.

[156] Gurobi Optimizer software, `http://www.gurobi.com/`, last visited: July 2011.

[157] Taylan, P., Özkurt, F.Y. and Weber, G.-W. *An approach to mean shift outlier model (MSOM) by Tikhonov regularization and conic programming*, Preprint **3**, Institute of Applied Mathematics, Middle East Technical University, (2010).

[158] Hibbs, M.A. *The effects of pre-processing and parameter choices on searches through large gene expression data collections*, 2009 IEEE International Workshop On Genomic Signal Processing and Statistics, 164-167, (2009).

[159] Ben-Tal, A. and Nemirovski, A. *Robust optimization - methodology and applications*, Mathematical Programming, **92(3)**, 453-480, (2002).

[160] Ben-Tal, A. and Nemirovski, A. *Robust convex optimization*, Mathematics of Operations Research, **23**, 769-805, (1998).

[161] Ben-Tal, A. and Nemirovski, A. *Robust solutions to uncertain linear programs*, Operations Research Letters, **25(1)**, 1-13, (1999).

[162] Ben-Tal, A., El-Ghaoui, L. and Nemirovski, A. *Robust semidefinite programming, in Semidefinite Programming and Applications*, R. Saigal, L. Vandenberghe, and H. Wolkowicz (Eds.), Kluwer Academic Publishers, Dordrecht, (2000).

[163] Ben-Tal, A., El-Ghaoui, L. and Nemirovski, A. *Robust Optimization*, Princeton University Press, New Jersey, (2009).

[164] Bertsimas, D. and Sim, M. *Price of robustness*, Operations Research, **52(1)**, 35-53, (2004).

[165] Bertsimas, D., Brown, D.B. and Caramanis, C. *Theory and applications of robust optimization*, Technical Report, University of Texas, Austin, TX, USA, (2007).

[166] Boni, O. *Robust solutions of conic quadratic problems*, Ph.D. Dissertation, Technion, Israeli Institute of Technology, (2007).

[167] El-Ghaoui, L. and Lebret, H. *Robust solutions to least-square problems to uncertain data matrices*, SIAM Journal on Matrix Analysis and Applications, **18**, 1035-1064, (1997).

[168] El-Ghaoui, L., Oustry, F. and Lebret, H. *Robust solutions to uncertain semidefinite programs*, SIAM Journal on Optimization, **9**, 33-52, (1998).

[169] Fabozzi, F.J., Kolm, P.N., Pachamanova, D.A. and Focardi, S.M. *Robust Portfolio Optimization and Management*, Wiley Finance, New Jersey, (2007).

[170] Werner, R. *Cascading: An adjusted exchange method for robust conic programming*, Central European Journal of Operations Research, **16**, 179-189, (2008).

[171] Kropat, E., Weber G.-W. and Rückmann, J. *Regression analysis for clusters in gene-environment networks based on ellipsoidal calculus and optimization*, to appear in Dynamics of Continuous, Discrete and Impulsive Systems, `http:monotone.uwaterloo.ca/~journal`.

[172] Bröcker, T. and Lander, L. *Differentiable Germs and Catastrophes*. London Mathematical Society Lecture Note Series **17**, Cambridge University Press, London, (1975).

[173] Özmen, A. and Weber, G.-W. *Robust conic generalized partial linear models using RCMARS method - A robustification of CGPLM*. Preprint **5** at Institute of Applied Mathematics, Middle East Technical University, (2011), Proceedings of Fifth Global Conference on Power Control and Optimization (June 1-3, 2011, Dubai), ISBN: 983-44483-49.

VITA

PERSONAL INFORMATION

Surname, Name: Defterli, Özlem

Nationality: Turkish (TC)

Date and Place of Birth: 6 October 1979, Ankara

Phone: +90 312 284 4500 Internal: 4037

email: defterli@cankaya.edu.tr

EDUCATION

Ph.D. Department of Mathematics, August 2011
Middle East Technical University-Ankara
Advisor: Assoc. Prof. Dr. Songül Kaya Merdan
Co-advisor: Prof. Dr. Gerhard-Wilhelm Weber
Thesis Title: Modern Mathematical Methods in Modeling and Dynamics of Regulatory Systems of Gene-Environment Networks

M.Sc. Department of Mathematics and Computer Science (with scholarship)
June 2004, Çankaya University-Ankara
Advisor: Assoc. Prof. Dr. Dumitru Baleanu
Thesis Title: Mathematical Aspects of Superintegrable Systems in Two Dimensions

B.S. (Minor) Department of Computer Engineering (with scholarship)
June 2002, Çankaya University-Ankara
B.S. (Major) Department of Mathematics and Computer Science (with scholarship)
June 2002, Çankaya University-Ankara

High School Alparslan Super High School-Ankara, June 1997

WORK EXPERIENCE

Year	Place	Enrollment
2008-	Çankaya University, Department of Mathematics and Computer Science	Instructor
2002-2008	Çankaya University, Department of Mathematics and Computer Science	Research Assistant

FOREIGN LANGUAGES

Turkish (Native), English (Advanced),

COMPUTER SKILLS

Program Languages: Visual C, FORTRAN, Visual Basic, JAVA,
Package Software: MATLAB, Maple, LATEX, MS Office, Microsoft FrontPage, Adobe Photoshop

AWARDS and GRANTS

Graduation as a *high honor with the first degree* from the Mathematics and Computer Science Department, and from the Faculty of Art and Sciences in Çankaya University, 2002

Being third finalist for the *FDA08 Young Mittag-Leffler Award*, 2008

TUBITAK National Scholarship for PhD Students, 2007-2009

PUBLICATIONS (in SCI)

1. O. Defterli, A. Fugenschuh, G.-W. Weber. *Modern tools for the time-discrete dynamics and optimization of gene-environment networks*, Communications in Nonlinear Science and Numerical Simulations, **16(12)**, 4768-4779, (2011).
2. A. Ozmen, G.-W. Weber, Z. Cavusoglu, O. Defterli. *The New Robust Conic GPLM Method - With an Application to Finance and Regulatory Systems: Prediction of Credit Default and a Process Version*, Journal of Global Optimization, (submitted), (2011).
3. G.-W. Weber, O. Defterli, S.Z. Alparslan Gok, E. Kropat. *Modeling, inference and optimization of regulatory networks based on time series data*, European Journal of Operational Research, **211(1)**, 1-14, (2011).
4. O.P. Agrawal, O. Defterli, D. Baleanu. *Fractional optimal control problems with several state and control variables*, Journal of Vibration and Control, **16(13)**, 1967-1976, (2010).
5. O. Defterli. *A numerical scheme for two-dimensional optimal control problems with memory effect*, Computers and Mathematics with Applications, **59(5)**, 1630-1636, (2010).
6. D., Baleanu, O. Defterli, O.P. Agrawal. *A central difference numerical scheme for fractional optimal control problems*, Journal of Vibration and Control, **15(4)**, 583-597, (2009).
7. O. Defterli, D. Baleanu. *Symplectic algorithm for systems with second-class constraints*, Czechoslovak Journal of Physics, **56(11)**, 1117-1122, (2006).
8. O. Defterli, D. Baleanu. *Projector quantization method of systems with linearly dependent constraints*, Czechoslovak Journal of Physics, **55(11)**, 1379 - 1384, (2005).
9. D. Baleanu, O. Defterli. *Killing-Yano tensors and angular momentum*, Czechoslovak Journal of Physics, **54(2)**, 157 - 166 , (2004).
10. O. Defterli, D. Baleanu. *Killing-Yano tensors and superintegrable systems*, Czechoslovak Journal of Physics, **54(11)**, 1215 - 1222, (2004).

CITATIONS

11 pure citations in SCI for the *Paper 6* in publication list

1 pure citation in SCI for the *Paper 5* in publication list

1 pure citation in SCI for the *Paper 10* in publication list

CONFERENCE PROCEEDINGS (as Refereed International)

(i) *Chapters in a Book:*

1. D. Baleanu, O. Defterli. *Killing-Yano tensors, surface terms and superintegrable systems*, Global Analysis and Applied Mathematics, AIP Conference Proceedings, Vol.**729**, Editors: K. Taş, D. Krupka, O. Krupkova, D. Baleanu, Springer-Verlag, ISBN 0-7354-0209-4, pp. 99-105, (2004), (in SCI).
2. O. Defterli, D.Baleanu. *Hidden symmetries of two dimensional superintegrable systems*, In the book: Mathematical Methods in Engineering, Editors: K.Tas, J.A.Tenreiro Machado, D.Baleanu. Springer, The Netherlands, ISBN-10 1-4020-5677-X (HB), pp. 159-166, (2007).

(ii) *Papers in CD:*

1. O. Defterli, D. Baleanu, Om P. Agrawal. Direct numerical scheme for fractional optimal control problems in multi dimensions, In: Proceedings of 8^{th} Portuguese Conference on Automatic Control (CONTROLO 2008), July 21-23, 2008, Vila Real, Portugal, Editor: Jose Boaventura Cunha. Publisher: Universidade de Tras-os-Montes e Alto Douro (UTAD), ISBN: 978-972-669-877-7, 279-284, (2008).
2. O. Defterli, A. Fugenschuh, G.W. Weber. *New discretization and optimization techniques with results in the dynamics of gene-environment networks*, In: Proceedings of the 3^{rd} Global Conference on Power Control and Optimization (PCO 2010), February 2-4, 2010, Gold Coast, Australia, Editors: N. Barsoum, P.Vasant, R. Habash, ISBN: 978-983-44483-1-8, (2010).

REFEREEING

Discrete and Continuous Dynamical Systems- Series A

Advances in Difference Equations

Differential Equations and Dynamical Systems

Communications in Nonlinear Science and Numerical Simulation

Physica A

Bifurcation and Chaos

Central European Journal of Physics

TCCT Conference Paper Management System

PRESENTATIONS IN INTERNATIONAL SCIENTIFIC MEETINGS

(i) Talks Presented:

1. *Modern Tools for the Discretization and Optimization of Dynamical Models for Gene-Environment Networks* - 3^{rd} International Conference on Nonlinear Science and Complexity - Ankara, Turkey, July 28 - 31, 2010.
2. Talks given through the *ERASMUS* activities: Series of seminars given in the Department of Mathematics of the University of Aveiro (Portugal) during the visit under the "Erasmus Teaching Staff Exchange Program" in the period of September 14 - 18, 2009.
3. *Two Dimensional Fractional Optimal Control Problem Using a Direct Numerical Scheme* - 3^{rd} International IFAC Workshop on Fractional Differentiation and its Applications - Ankara, Turkey, November 05 -07, 2008.
4. *Direct Numerical Scheme for Fractional Optimal Control Problems in Multi Dimensions* - 8^{th}Portuguese Conference on Automatic Control - Vila Real, Portugal, July 21-23, 2008.
5. *Chain by Chain Method and Projector Quantization Approach for Systems with Second-Class Constraints* - 15^{th} International Colloquium on Integrable Systems and Quantum Symmetries - Prague, Czech Republic, June 15 - 17, 2006.
6. *Hidden Symmetries of Two Dimensional Superintegrable Systems* - Mathematical Methods in Engineering, International Symposium - Ankara, Turkey, April 27 - 29, 2006.
7. *Projector Quantization Method of Systems with Linearly Dependent Constraints* - 14^{th}International Colloquium on Integrable Systems and Quantum Groups - Prague, Czech Republic, June 16 - 18, 2005.
8. *Geometrical Aspects of Superintegrability in Two Dimensional Space of Non-Constant Curvature* - 3^{rd} Geometry Symposium - Eskisehir, Turkey, July 4 - July 6, 2005. (National conference)
9. *Killing - Yano Tensors and Superintegrable Systems* - 13^{th} International Colloquium - Integrable Systems and Quantum Groups - Prague, Czech Republic, June 17 - 19, 2004.
10. *About Killing -Yano Tensors, Superintegrable Systems and Surface Terms* - International Workshop on Global Analysis - Ankara, Turkey, April 15 - 17, 2004.
11. *Geometrical Aspects of the Surface Terms*, (Poster) - 8^{th}International Summer School in Global Analysis and Applications (Geometrical Structures in the Calculus of Variations) - Brno, Czech Republic, August 4 - 8, 2003.

(ii) Talks Co-Authed:

1. G.-W. Weber, O. Defterli, L. Özdamar, C.S. Pedamallu, B. Temoçin, Y. Yildiz, *Recent Advances in Mathematical Prediction of Dynamics under Different Assumptions on Time and Uncertainty*, 23^{th} International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany, August 1-5, 2011.
2. G.-W. Weber, A. Özmen, Z. Çavuşoğlu, O. Defterli, *The New Robust Conic GPLM Method with an Application to Finance and Regulatory Systems: Prediction of Credit Default and a Process Version*, 9^{th} EUROPT Workshop on Advances in Continuous Optimization, Ballarat, Australia, July 8-9, 2011.
3. G.-W.Weber, S.Z. Alparslan-Gök, E. Kropat, O. Defterli, F. Yerlikaya Özkurt, A. Fügenschuh, *Research Progress and New Applications of Eco-Finance Networks and Cooperative Game Theory*, International Seminar on Operational Research, Medan, Indonesia, July 27-28, 2011.

4. G.-W. Weber, A. Özmen, Z. Çavuşoğlu, O. Defterli, *Prediction of Default Probabilities in Emerging Markets and of Dynamical Regulatory Networks by New Robust Conic GPLMs and Their Optimization*, International Seminar on Operational Research, Medan, Indonesia, July 27-28, 2011.
5. O. Defterli, G.-W. Weber, E. Kropat, S.Z. Aparslan Gök and A. Fügenschuh, *Modeling, Inference and Optimization of Regulatory Networks Based on Time Series Data*, OR 2010 - International Conference on Operations Research, Munich, September 1-3, 2010 .
6. G.-W. Weber, O. Defterli and A. Fügenschuh, *New Advances in Prediction of Gene-Environment Networks by Applied Mathematics Tools*, 5^{th} International Summer School Achievements and Applications of Contemporary Informatics, Mathematics and Physics, National University of Technology of the Ukraine, Kiev, Ukraine, August 3-15, 2010.

PARTICIPATION IN INTERNATIONAL SCIENTIFIC MEETINGS

1. 3^{rd} International Conference on Nonlinear Science and Complexity (NSC'10), Ankara, Turkey, July 28 - 31, 2010.
2. International Conference on New Trends in Nanotechnology and Nonlinear Dynamical Systems (NNDS'10), Ankara, Turkey, July 25 - 27, 2010.
3. 3^{rd} International IFAC Workshop on Fractional Differentiation and its Applications (FDA'08), Ankara, Turkey, November 05 - 07, 2008.
4. International Workshop on New Trends in Science and Technology (NTST'08), Ankara, Turkey, November 03 - 04, 2008.
5. 8^{th} Portuguese Conference on Automatic Control (CONTROLO'08), Vila Real, Portugal, July 21 - 23, 2008.
6. Mathematical Methods in Engineering (MME'06), International Symposium, Ankara, Turkey, April 27 - 29, 2006.
7. The Jubilee 15^{th} International Colloquium on Integrable Systems and Quantum Symmetries (ISQS-15), Prague, Czech Republic, June 15 - 17, 2006.
8. 14^{th} International Colloquium on Integrable Systems, Prague, Czech Republic, June 16 - 18, 2005.
9. 13^{th} International Colloquium - Integrable Systems and Quantum Groups, Prague, Czech Republic, June 17 - 19, 2004.
10. International Workshop on Global Analysis (IWGA), Ankara, Turkey, April 15 - 17, 2004.
11. Geometrical Structures in the Calculus of Variations - 8^{th} International Summer School in Global Analysis and Applications, Brno, Czech Republic, August 4 - 8, 2003.
12. NATO ASI programme on Computational Noncommutative Algebra with Applications, Tuscany, Italy, July 6 - 19, 2003.

PARTICIPATION IN NATIONAL SCIENTIFIC MEETINGS

1. 6^{th} Ankara Mathematics Days, Hacettepe University, Ankara, Turkey, June 02-03, 2011.
2. 5^{th} Ankara Mathematics Days, TOBB ETU, Ankara, Turkey, June 03-04, 2010.
3. 4^{th} Ankara Mathematics Days, METU, Ankara, Turkey, June 04-05, 2009.
4. 3^{rd} Geometry Symposium, Osmangazi University, Eskisehir, Turkey, July 04-06 July 6, 2005.
5. 2^{nd} Geometry Symposium, Sakarya University, Turkey, June 30-July 03, 2004.

CONFERENCE ORGANIZATION

1. Member of the Organizing Committee of the 3rd *International IFAC Workshop on Fractional Differentiation and its Applications -FDA'08*, Ankara-Turkey, November 05-07, 2008.
2. Member of the Organizing Committee of the *New Trends in Nanotechnology and Nonlinear Dynamical Systems - NNDS 2010*, Ankara, Turkey, July 25 - 27, 2010.
3. Member of the Organizing Committee of the 3rd *International Conference on Nonlinear Science and Complexity - NSC 2010* - Ankara, Turkey, July 28 - 31, 2010.
4. Special Symposia Organizer, Parallel Session Chair: in the 3rd *International Conference on Nonlinear Science and Complexity - NSC 2010* - Ankara, Turkey, July 28 - 31, 2010.