Norges miljø- og biovitenskapelige universitet

Master Thesis 2016
30 credits
Department of Mathematical Sciences and Technology

Metamodels of Gene Regulatory Networks

Metamodeller av genregulatoriske nettverk

Linda Mei Brandett Thinn
Master of Science - Environmental Physics and Renewable Energy

Acknowledgements

This thesis completes my master's degree in Environmental Physics and Renewable Energy at the Norwegian University of Life Sciences (NMBU).

I would like to thank my supervisor Prof. Arkadi Ponossov, who kept me focused when I was too eager, and gave me advice when I was confused. I greatly appreciate our weekly Wednesday meetings; thank you for sharing ideas, knowledge and enthusiasm for the subject.

I would also like to thank my parents for always supporting me.

Ås, 15 December 2016

Linda Mei Brandett Thinn

Abstract

The Hill function is a sigmoid commonly used in modelling of genetic networks. This thesis presents genetic networks described by systems of differential equations. By principal component analysis - a procedure that reduces the dimensionality of a system - a metamodel of the Hill function is established. The metamodel is evaluated through studies of genetic networks of different order. Further, extreme value problems, regarding determination of the production terms of these networks, are considered.

In a generic situation it was proved that extrema are obtained in the corner points. The metamodel turned out to be a good fit, especially considering that known benchmark problems were implemented. However, deviations close to the limits of the metamodel were discovered; in this region the model should not be applied uncritically.

Sammendrag

Hill-funksjonen er et sigmoid som er mye brukt i modellering av genetiske nettverk. I denne oppgaven presenteres genetiske nettverk beskrevet av differensialligninger. En metamodell for Hill-funksjonen er laget ved å bruke prinsipalkomponentanalyse - en teknikk som reduserer systemets dimensjonalitet. Metamodellen evalueres ved å studere genetiske nettverk av forskjellig orden. I forbindelse med bestemmelsen av produksjonsledd for disse nettverkene, blir ekstremalverdiproblemer tatt i betraktning.

I det generiske tilfellet ble det bevist at ekstremalverdiene oppnås i hjørnepunktene. Det er rimelig å anse metamodellen som en god approksimasjon, spesielt med tanke på at det var mulig å implementere kjente benchmarkproblemer. Videre ble avvik nær modellens grenser oppdaget; den bør ikke anvendes ukritisk i dette området.

Notation

	Notation
Singular Value Decomposition	SVD
Principal Component Analysis	PCA
Concentration	x
Steepness parameter	q
Threshold value	θ

Contents

1 Introduction **1**

2 Gene Regulatory Networks **3**
2.1 Biological Context 3
2.2 Mathematical Framework 4
2.3 Sliding Mode 5

3 Mathematical Theory **9**
3.1 The Singular Value Decomposition 9
3.2 The Principal Component Analysis 11
3.3 The Tensor Product 12
3.4 The Condition Number 13
3.5 Interpolation 14
3.6 The Runge-Kutta Method 15
3.7 The Relative Squared Error 17
3.8 Extreme Values of Functions of Two Variables 18
3.9 Extreme Values of Functions of n Variables 19
3.10 Mathematical Induction Principle 21

4 The Hill Function **23**
4.1 The Hill Function 23
4.2 The Metamodel 24

5 The Scalar Case **29**
5.1 A Genetic Network of One Gene 29
5.2 The Metamodel 29

6 2×2 System of Genes **33**
6.1 A Genetic Network of Two Genes 33
6.2 Analytic Determination of the Production Term 33
6.3 The Metamodel 34

6.4 A Benchmark Problem 37

7 Modelling Circadian Oscillations 39

8 3×3 System of Genes 45
8.1 A Genetic Network of Three Genes 45
8.2 The Metamodel 46
8.3 Sources of Error in the Metamodel 48

9 Determination of the Production Term for a $n \times n$ System of Genes 53

10 Discussion 57

11 Conclusion and Outlook 59

Appendix A Matlab Codes 63
A.1 Approximating the Hill Function 63
A.2 Solving Systems of Differential Equations 65

Chapter 1

Introduction

"A theory has only the alternative of being right or wrong. A model has a third possibility: it may be right, but irrelevant [1]*,"* Manfred Eigen [11].

Physics, economics or biology - mathematical models have an endless amount of applications. We want to describe systems. We strive to understand, explain and predict. And in order to do so, mathematical models are developed. However, these models can be complex and complicated. It may be convenient to work with something less extensive. We want to simplify without reducing the reliability; this is the purpose of the metamodel.

In 1965, B. C. Goodwin proposed a mathematical model describing a gene regulatory network with differential equations. Strong threshold and switching effects are common in such systems, and this was represented by the Hill function.

In the PhD thesis of Julia Isaeva, metamodels of 38 line curvature, among them sigmoids, were established by principal component analysis. The Hill function was among the best fits. Further, a system of differential equations including these metamodels was suggested [4].

The aim of this thesis is to implement and analyse metamodels of typical gene regulatory networks. A discrete approximation of the Hill function is yielded using principal component analysis. This is the basis of the metamodels later presented. Metamodels of genetic networks of one, two and three genes are constructed; the accuracy is studied in order to determine whether they are satisfactory. In addition, extreme value problems regarding determination of the production terms are considered.

Benchmark problems help validate and verify new implementations. Thus,

[1] These are the words of the biophysical chemist Manfred Eigen. He was, together with Ronald George Wreyford Norrish and George Porter, awarded with the Nobel Prize in Chemistry in 1967 for their studies of extremely fast chemical reactions [1].

for further evaluation two benchmark problems are solved. Thereby it is possible to compare the metamodel to a reference solution.

The main advantage of these metamodels is that principal component analysis reduces the dimensionality of the system. With a big network, this has great impact on the efficiency.

Chapter 2

Gene Regulatory Networks

2.1 Biological Context

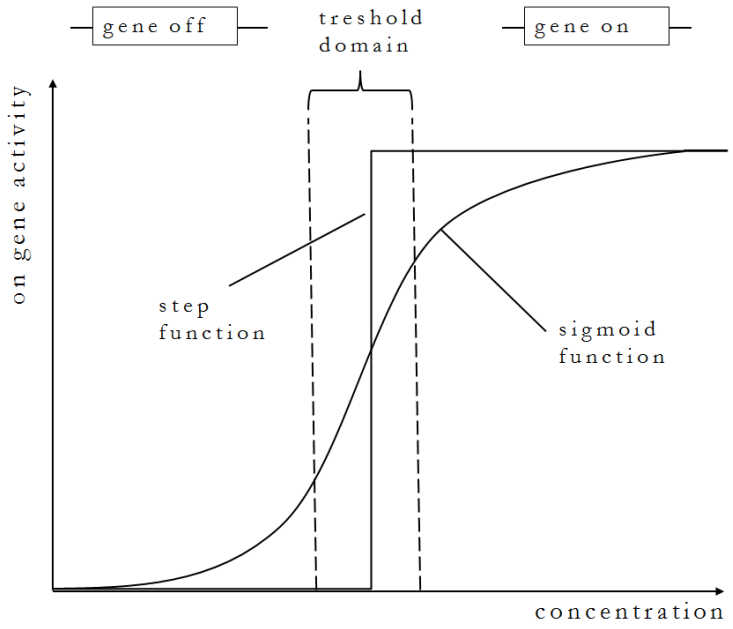


Figure 2.1: The relation between threshold and gene activity. Genes are switched "on" or "off" depending on whether the concentration of a certain regulatory factor is below or above the threshold [12].

The purpose of gene regulatory models is to describe and illustrate the inter-

actions of genes; how the concentration of some genes affect the production or degradation of others.

In a wide range of biological phenomena, interactions with strong threshold and switching effects appear. In gene regulatory networks they are frequent. This behaviour is commonly described by sigmoid functions or step functions. There are few general guidelines for choosing of sigmoids. Often sigmoid curves are preferable to step functions in biological models; the region around the threshold is in many circumstances important, and step functions are discontinuous at this point [17].

However, the threshold domains in gene regulatory networks are often narrow. Therefore it is not unreasonable to think of such a phenomena as a binary on-off device. See figure 2.1.

Among the more used sigmoid functions, is the Hill function. It was first introduced in 1910. The biochemist A. V. Hill formulated the equation in order to explain the binding of oxygen to hemoglobin [19].

The same function played, 55 years later, a central role when Brian C. Goodwin proposed a new mathematical model describing a gene regulatory network with differential equations. The Hill function represented a biological threshold effect in the nucleus. [12].

2.2 Mathematical Framework

Before describing the genetic network, we first establish the most elementary model. The scalar case: a one gene network - an isolated gene in a population. The rate of change in a gene concentration is defined as the difference between the production rate and the degradation rate

$$\dot{x} = P - Dx$$

where $P \geq 0$ is the production rate, $D \geq 0$ is the degradation rate and $x(t)$ is the gene concentration. This can easily be expanded to a genetic network of N genes

$$\dot{x_i} = P_i - D_i x_i, \qquad i = 1, 2, ..., N$$

Consider a biological system with strong threshold mechanisms, as illustrated in figure 2.2. Protein x_1 and x_2 affects each other, while protein x_3 rely on both x_1, x_2 and itself. Their production or degradation rate only changes when certain thresholds are crossed. The rates depend on sigmoid functions z_i of different steepness [13, 18]. An interaction structure such as

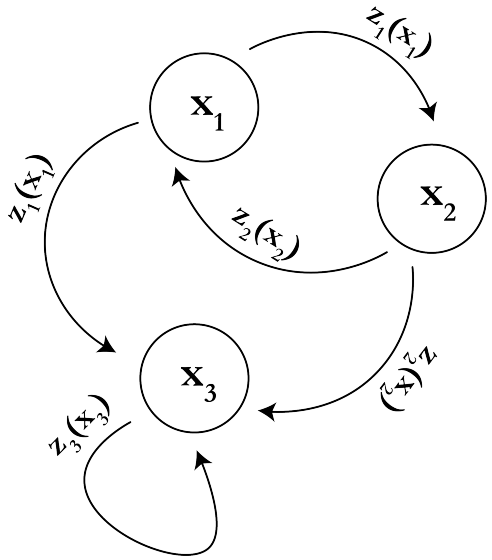


Figure 2.2: Interaction diagram of a population of three genes [18].

this can be described

$$\dot{x_1} = P_1(z_2) - D_1(z_2)x_1$$

$$\dot{x_2} = P_2(z_1) - D_2(z_1)x_2$$

$$\dot{x_3} = P_3(z_1, z_2, z_3) - D_1(z_1, z_2, z_3)x_3$$

2.3 Sliding Mode

A not unknown phenomena in gene regulatory networks is sliding mode. When a system tends to «slide»along a boundary (e.g., surface, line) after a certain time t_r and stay there thereafter, the system is said to be in sliding mode when $t > t_r$ [21]. See figure 2.3. This is further illustrated by studying the behaviour of the solutions in the phase plane. See figure 2.4.

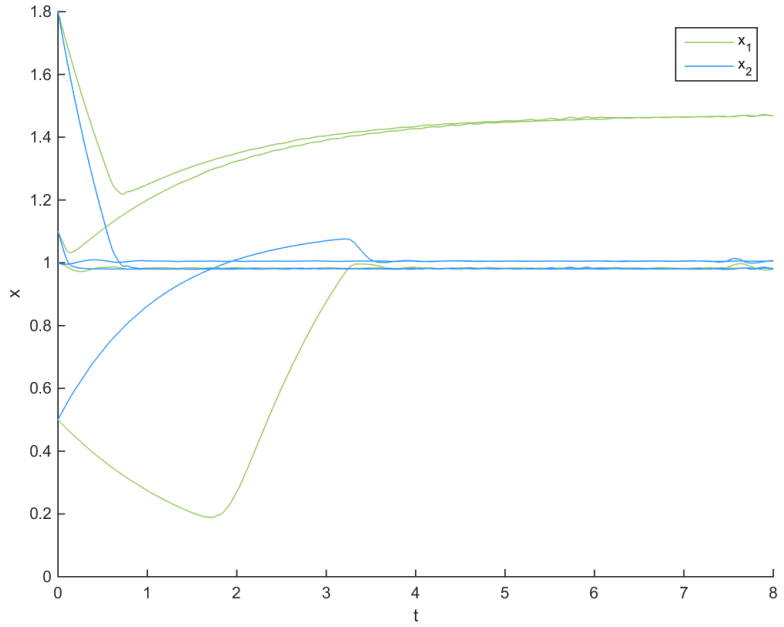


Figure 2.3: Illustration of a system where both x_1 and x_2 are in sliding mode [16].

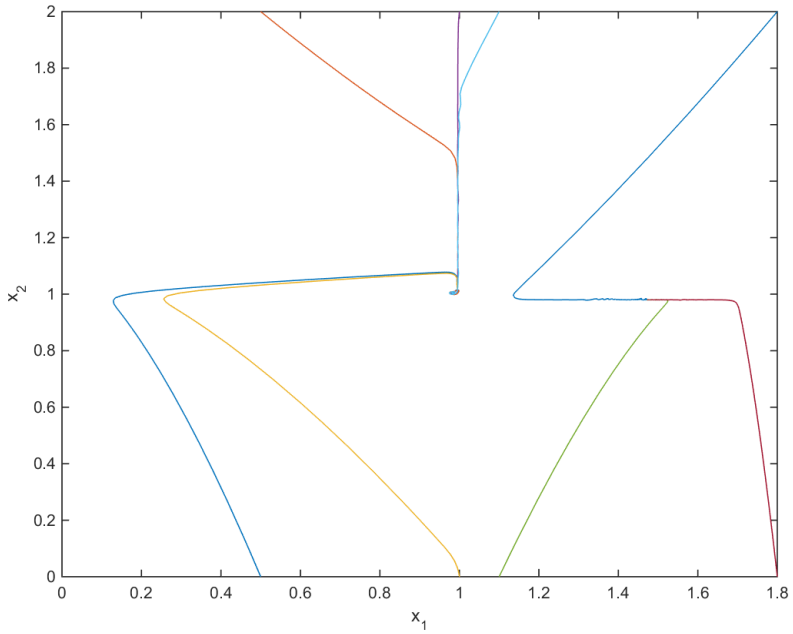


Figure 2.4: Trajectories in the phase plane corresponding to the solutions illustrated in figure 2.3 [16].

In a genetic network of the type previously outlined, such behaviour would indicate that the concentration of one or more genes stabilizes. The reason for this, is usually a feedback process; a positive feedback amplifies an effect, while a negative feedback reduces it. Autoregulation is a feedback process where e.g a gene regulates its own production[16]. It may cause mathematical problems because the derivative $\frac{dx}{dt}$ quickly approaches zero.

Chapter 3

Mathematical Theory

3.1 The Singular Value Decomposition

Singular value decomposition (SVD) is a factorization of matrices. The technique is commonly used, and among it's many applications we find image analysis, statistics and data processing.

Let A be a $n \times m$ matrix. Then the singular value decomposition of A is defined as $A = USV^\tau$. S is a $n \times m$ matrix with nonnegative diagonal entries, U is a $n \times n$ orthonormal matrix and V is a $m \times m$ orthonormal matrix.

The singular values, $\sigma_1, \sigma_2, ..., \sigma_n$, of the matrix A, are the positive square roots of the eigenvalues of the associated Gram matrix $G = A^\tau A$. Thus, $\sigma_1^2, \sigma_2^2, ..., \sigma_n^2$ are the eigenvalues of G. The singular values are arranged in descending order: $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n > 0$. If the matrix A is square, then so is S. For $n = m$ then

$$S_{n\times n} = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & 0 \\ & & \sigma_3 & \\ & 0 & & \ddots \\ & & & & \sigma_n \end{pmatrix}$$

If $n > m$ then

$$S_{n\times m} = \begin{pmatrix} \sigma_1 & & & & \\ & \sigma_2 & & 0 & \\ & & \sigma_3 & & \\ & 0 & & \ddots & \\ & & & & \sigma_m \\ 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix} \Big\} n - m$$

If $m > n$ then

$$S_{n\times m} = \begin{pmatrix} \sigma_1 & & & & 0 & 0 \\ & \sigma_2 & & 0 & \vdots & \vdots \\ & & \sigma_3 & & \vdots & \vdots \\ & 0 & & \ddots & \vdots & \vdots \\ & & & & \sigma_n & 0 \quad 0 \end{pmatrix} \underbrace{\qquad}_{m-n}$$

U and V are both orthonormal matrices. The columns are called, respectively, left-singular and right-singular vectors of A.

$$V_{m\times m} = \begin{pmatrix} v_1 & v_2 & ... & v_m \end{pmatrix} \quad and \quad U_{n\times n} = \begin{pmatrix} u_1 & u_2 & ... & u_n \end{pmatrix}$$

The right-singular vectors of A, v_i, are the unit eigenvectors, $\|v_i\| = 1$, corresponding to the eigenvalues ${\sigma_i}^2$. By the following relation, the columns of U can be constructed

$$u_i = \frac{1}{\sigma_i} A v_i, \qquad i = 1, 2, ..., n, \qquad \|u_i\| = 1 \tag{3.1}$$

where A is a $n \times m$ matrix, σ_i are the singular values of A and v_i are the corresponding eigenvectors.

The singular value decomposition of matrix A can also be written on vector based form

$$A = \sigma_1 u_1 v_1^\tau + \sigma_2 u_2 v_2^\tau + ... + \sigma_p u_p v_p^\tau \tag{3.2}$$

where A is a $n \times m$ matrix, σ_i are the singular values, u_i are the left-singular vectors and v_i are the right-singular vectors. $i = 1, 2, ..., p$, and $p = min\{m, n\}$ [8, 18].

3.2 The Principal Component Analysis

Principal component analysis (PCA) is a method frequently used to reduce the complexity of a system. It is useful for suppressing redundant information and extracting relevant data. We seek the principal components with the largest variance, while those with lower variance are considered noise.

Singular value decomposition is closely related to PCA. With SVD the matrix A can be represented by vectors (3.2), and the principal components expressed as «$\sigma_i u_i v_i^\tau$». Then, «$\sigma_1 u_1 v_1^\tau$» is the first principal component, «$\sigma_2 u_2 v_2^\tau$» the second principal component, etc. In order to determine the number of components k needed to achieve a satisfactory model, we consider all components with index $i > k$ as error, e_i [18].

$$
\begin{aligned}
A = & \underbrace{\sigma_1 u_1 v_1^\tau}_{\text{first component}} + e_1, \quad \|e_1\| = \sigma_2 \\
A = & \underbrace{\sigma_1 u_1 v_1^\tau}_{\text{first component}} + \underbrace{\sigma_2 u_2 v_2^\tau}_{\text{second component}} + e_2, \quad \|e_2\| = \sigma_3 \\
& \vdots \\
A = & \sigma_1 u_1 v_1^\tau + ... + \sigma_k u_k v_k^\tau + e_k, \quad \|e_k\| = \sigma_{k+1}
\end{aligned}
$$

This can also be expressed as

$$
\begin{aligned}
A &= \sum_{i=1}^{k} t_i p_i^\tau + e_k, \quad \|e_k\| = \sigma_{k+1} \\
t_i &= \sigma_i u_i, \quad \|t_i\| = \sigma_i \\
p_i &= v_i, \quad \|p_i\| = 1
\end{aligned}
\tag{3.3}
$$

where A is a $n \times m$ matrix, t_i are the scores, p_i are the loadings, σ_i are the singular values, u_i are the left-singular vectors, v_i are the right-singular vectors, e_k is the error and k is the number of principal components [18].

3.3 The Tensor Product

Let A be a $n \times m$ matrix and B be a $p \times q$ matrix. Then the tensor product of A and B is defined as the $mp \times nq$ matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nm}B \end{pmatrix}$$

Consider the matrix $C = A \otimes B$ with the singular value decomposition $C = USV^\tau$. Given the decompositions

$$A = U_A S_A V_A^\tau \quad and \quad B = U_B S_B V_B^\tau$$

then

$$\begin{aligned} C &= (U_A S_A V_A^\tau) \otimes (U_B S_B V_B^\tau) \\ &= (U_A \otimes U_B)(S_A \otimes S_B)(V_A^\tau \otimes V_B^\tau) \\ &= USV^\tau \end{aligned} \tag{3.4}$$

where $C = A \otimes B$ and U, S, V are the matrices from the SVD of the $mp \times nq$ matrix C, the $n \times m$ matrix A and the $p \times q$ matrix B [7].

Or, written less formal: $SVD(A \otimes B) = SVD(A) \otimes SVD(B)$. To illustrate the concept, assume we have two functions of two independent variables

$$f_1(x_1, y_1) \quad and \quad f_2(x_2, y_2)$$

We discretize the functions over intervals containing N points

$$\begin{aligned} x_1 &= x_1^1, x_1^2, ..., x_1^N, \qquad & y_1 &= y_1^1, y_1^2, ..., y_1^N \\ x_2 &= x_2^1, x_2^2, ..., x_2^N, \qquad & y_2 &= y_2^1, y_2^2, ..., y_2^N \end{aligned}$$

For simplicity, let $N = 2$. Then the functions can be expressed discreetly as the 2×2 matrices F_1 and F_2.

$$f_1(x_1, y_1) \to F_1 = \begin{pmatrix} f_1(x_1^1, y_1^1) & f_1(x_1^2, y_1^1) \\ f_1(x_1^1, y_1^2) & f_1(x_1^2, y_1^2) \end{pmatrix}$$

$$f_2(x_2, y_2) \to F_2 = \begin{pmatrix} f_2(x_2^1, y_2^1) & f_2(x_2^2, y_2^1) \\ f_2(x_2^1, y_2^2) & f_2(x_2^2, y_2^2) \end{pmatrix}$$

The product of the functions f_1 and f_2 will then be the 4×4 matrix F

$$f_1(x_1, y_1)f_2(x_2, y_2) \rightarrow F = \begin{pmatrix} f_1(x_1^1, y_1^1)F_2 & f_1(x_1^2, y_1^1)F_2 \\ f_1(x_1^1, y_1^2)F_2 & f_1(x_1^2, y_1^2)F_2 \end{pmatrix}$$

$$= F_1 \otimes F_2$$

With singular value decomposition and principal component analysis, the matrices can be approximated on the form (3.3)

$$\hat{F} = \sum_{i=1}^{k} t_i p_i^{\tau}, \qquad \hat{F_1} = \sum_{i=1}^{k} t_i^1, p_i^{1\tau}, \qquad \hat{F_2} = \sum_{i=1}^{k} t_i^2 p_i^{2\tau}$$

From equation (3.4) it follows that

$$\hat{F} = \hat{F_1} \otimes \hat{F_2}$$

For application issues, this means that the PCA of F_1 and F_2 can be used also when working with the product of the two functions f_1 and f_2. Notice that the scores t_i and the loadings p_i for the matrix F therefore does not need to be calculated.

3.4 The Condition Number

The condition number describes the error propagation of a system; a problem whose condition number is low is well-conditioned. A high condition number indicates ill-conditioning. A system is said to be well-conditioned if small changes in the data produce small changes in the solution. If, on the other hand, small changes in the data may cause large changes in the solution, the problem is considered ill-conditioned [6]. The condition number κ_p is defined

$$\kappa_p(A) = \|A\|_p \|A^{-1}\|_p \tag{3.5}$$

where A is a $n \times m$ matrix and p is the p-norm.

If $p = 2$ then

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_{max}}{\sigma_{min}} \tag{3.6}$$

where A is a $n \times n$ matrix, $\sigma_{max} = max\{\sigma_1, ..., \sigma_n\}$ is the largest singular value and $\sigma_{min} = min\{\sigma_1, ..., \sigma_n\}$ is the smallest singular value [18].

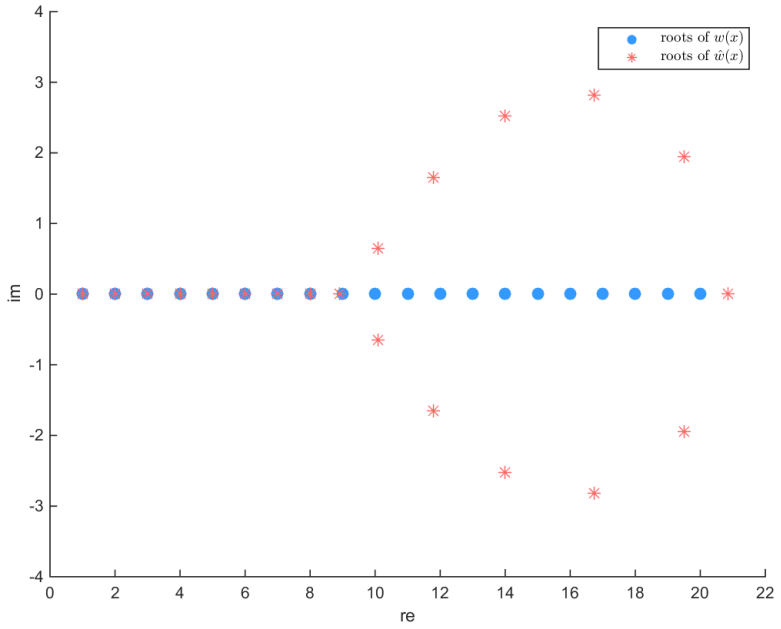


Figure 3.1: The roots of Wilkinson's polynomial $w(x)$ and $\hat{w}(x)$.

Consider the problem of finding the roots of Wilkinson's polynomial

$$\begin{aligned} w(x) = \prod_{i=1}^{20}(x-i) &= (x-1)(x-2)...(x-20) \\ &= x^{20} - 210x^{19} + ... + 20! \end{aligned}$$

On factorized form, finding the roots is trivial. Let us decrease the coefficient of x^{19} by 2^{-23}

$$\hat{w}(x) = x^{20} - (210 + 2^{-23})x^{19} + ... + 20!$$

This very small change, has a great impact on the roots. Some of the roots are displaced and, in fact, half of them are now complex. See figure 3.1. The reason for this behaviour, is the polynomial's huge condition number [18].

3.5 Interpolation

Given the discrete points $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$, it is possible to approximate a continuous function $y(x)$ for values between the given values of x. The technique is called interpolation and the points $x_0, x_1, ..., x_n$ nodes [6].

There are several kinds of interpolation (e.g., linear, polynomial, quadratic). In Matlab's built in function *interp1*, linear interpolation is the default method [9]. Linear interpolation is interpolation by straight lines. The precision of the interpolation will very much depend on the number of nodes n. As figure 3.2 illustrates, more data points increase the accuracy.

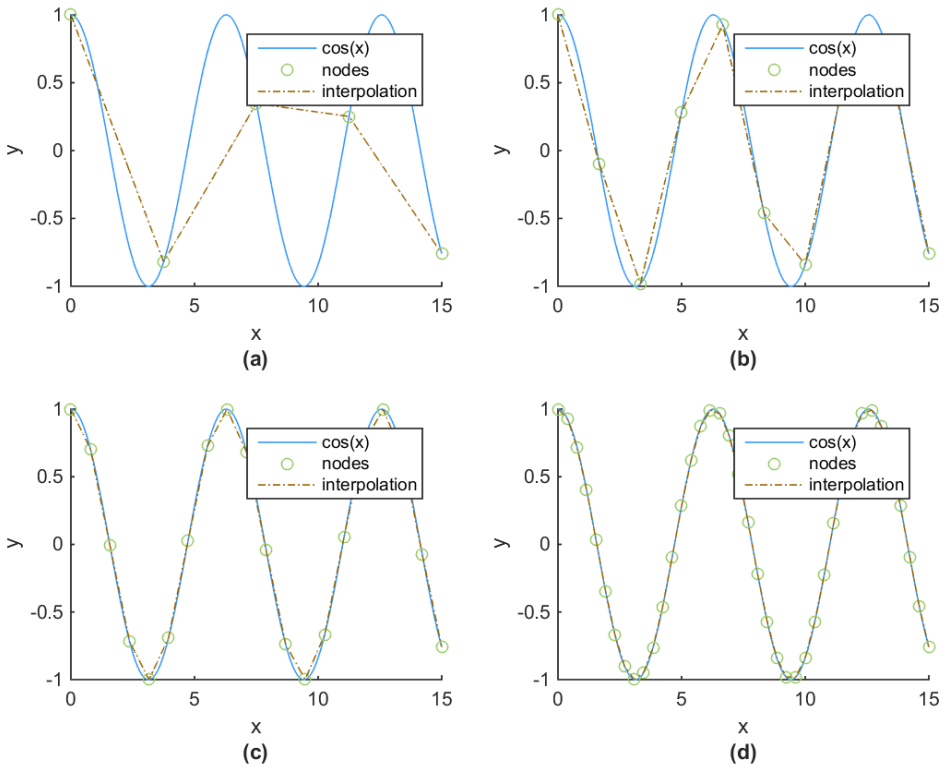


Figure 3.2: Interpolation of $y = cos(x)$ with n nodes. **(a)** $n = 5$ **(b)** $n = 10$ **(c)** $n = 20$ **(d)** $n = 40$

3.6 The Runge-Kutta Method

The Runge-Kutta method is an iterative, numerical solution procedure for solving first order differential equations. The method is a generalization of Simpson's rule; a rule for approximating the definite integrals by using parabolas.

Consider the scalar initial value problem

$$\dot{y} = f(t, y(t)), \qquad f(t_0) = y_0$$

The solution $y(t)$ is approximated at N discrete points

$$t_k = t_0 + kh, \qquad h = t_{k+1} - t_k$$

$$y(t_k) = y_k, \qquad k = 0, 1, .., N$$

Let us first introduce the integral

$$\int_{t_k}^{t_{k+1}} f(s, y(s))ds = y_{k+1} - y_k$$

Using Simpson's rule for numerical integration yields

$$\int_{t_k}^{t_{k+1}} f(s, y(s))ds = \frac{h}{6}\big(f(t_k, y_k) + 4f(t_{\frac{1}{2}}, y_{\frac{1}{2}}), + f(t_{k+1}, y_{k+1})\big)$$

where $t_{\frac{1}{2}}$ is the midpoint at $t_k + 0.5h$. At this stage y_{k+1} and $y_{\frac{1}{2}}$ are unknown, and are therefore approximated by the equation for the tangent line

$$y_{k+1} \approx y_k + hy_k$$

$$y_{\frac{1}{2}} \approx y_k + 0.5hy_k$$

Then the expression for y_{k+1} is

$$y_{k+1} = y_k + \frac{h}{6}[K_1 + 2K_2 + 2K_3 + K_2]$$

where

$$\begin{aligned} K_1 &= f(t_k, y_k) \\ K_2 &= f(t_k + \frac{1}{2}h, y_k + \frac{1}{2}hK_1) \\ K_3 &= f(t_k + \frac{1}{2}h, y_k + \frac{1}{2}hK_2) \\ K_4 &= f(t_k + h, y_k + hK_3) \end{aligned}$$

Notice that $f(t_{\frac{1}{2}}, y_{\frac{1}{2}}) = \frac{K_2 + K_3}{2}$. This scheme is called the fourth-order Runge-Kutta method, and is the basis of Matlab's built in function *ode45*[5, 18, 10].

Up until now we have considered the scalar initial value problem. The Runge-Kutta method can, however, easily be extended to first order systems [8]. For a system of m differential equations

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}(t)), \qquad \mathbf{f}(t_0) = \mathbf{y}_0$$

where

$$\mathbf{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{pmatrix}, \quad \mathbf{y}_0 = \begin{pmatrix} y_0^{(1)} \\ y_0^{(2)} \\ \vdots \\ y_0^{(m)} \end{pmatrix}, \quad \mathbf{f}(t, \mathbf{y}(t)) = \begin{pmatrix} f_1(t, \mathbf{y}) \\ f_2(t, \mathbf{y}) \\ \vdots \\ f_m(t, \mathbf{y}) \end{pmatrix}$$

Then, for a system

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h}{6}[\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_2]$$

where

$$\begin{aligned} \mathbf{K}_1 &= \mathbf{f}(t_k, y_k) \\ \mathbf{K}_2 &= \mathbf{f}(t_k + \frac{1}{2}h, \mathbf{y}_k + \frac{1}{2}h\mathbf{K}_1) \\ \mathbf{K}_3 &= \mathbf{f}(t_k + \frac{1}{2}h, \mathbf{y}_k + \frac{1}{2}h\mathbf{K}_2) \\ \mathbf{K}_4 &= \mathbf{f}(t_k + h, \mathbf{y}_k + h\mathbf{K}_3) \end{aligned}$$

3.7 The Relative Squared Error

From statistics, the coefficient of determination R^2 is a measure of goodness of fit; the proportion of total variability explained by the model. $R^2 \in [0, 1]$, where 1 is a perfect fit [14]. The coefficient of determination is defined

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{x}_i - x_i)^2}{\sum_{i=1}^{N}(\bar{x} - x_i)^2}, \tag{3.7}$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

where $\hat{x}_i$ are the predicted values, x_i are the observed values, $\bar{x}$ is the average of the observed values and N is the number of samples.

The relative squared error ζ^2 can then be expressed

$$\zeta^2 = 1 - R^2 \tag{3.8}$$

where R^2 is the coefficient of determination.

3.8 Extreme Values of Functions of Two Variables

In closed bounded regions, continuous functions of two variables assume extreme values. If $f(x, y)$ is a two variable real-valued function whose domain D is both closed and bounded, then f attains both an absolute maximum value M and an absolute minimum value m in D. That is

$$m \leq f(x, y) \leq M, \quad \forall x, y \in D$$

$$f(x_1, y_1) = m \quad and \quad f(x_2, y_2) = M$$

The only points where extreme values of the function $f(x, y)$ can be found, are critical interior points and boundary points. At critical points $\frac{\partial f}{\partial x}(a, b) = \frac{\partial f}{\partial y}(a, b) = 0$ or one or both of the partial derivatives fail to exist. The second partial derivatives test classifies the points [23].

i) *If* $\frac{\partial^2 f}{\partial x^2} > 0$ *and* $\frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - (\frac{\partial^2 f}{\partial x \partial y})^2 > 0$ *at (a,b), the point is a local minimum.*

ii) *If* $\frac{\partial^2 f}{\partial x^2} < 0$ *and* $\frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - (\frac{\partial^2 f}{\partial x \partial y})^2 > 0$ *at (a,b), the point is a local maximum.*

iii) *If* $\frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - (\frac{\partial^2 f}{\partial x \partial y})^2 < 0$ *at (a,b), the point is a saddle point.*

iv) *If* $\frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - (\frac{\partial^2 f}{\partial x \partial y})^2 = 0$, *the test is inconclusive.*

Figure 3.3 illustrates the function

$$g(x, y) = y^2 - x^2, \qquad x, y \in [-1, 1]$$

with the partial derivatives

$$\frac{\partial g}{\partial x} = -2x \quad and \quad \frac{\partial g}{\partial y} = 2y$$

Hence, local extrema can only occur at the origin (0,0). This is the only

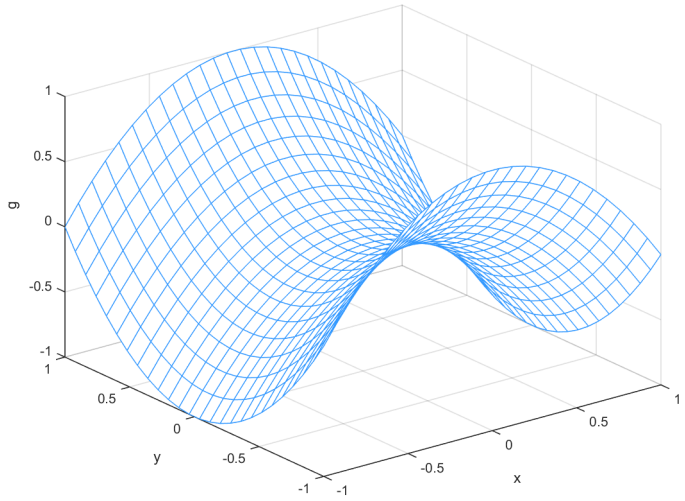


Figure 3.3: The graph of the function $g(x, y) = y^2 - x^2$

critical interior point and it is a saddle point. Investigating the boundaries

$$g(-1, y) = g(1, y) = y^2 - 1$$
$$g(x, -1) = g(x, 1) = 1 - x^2$$

$$\frac{\partial g}{\partial y}(-1, y) = \frac{\partial g}{\partial y}(1, y) = 2y$$
$$\frac{\partial g}{\partial x}(-1, x) = \frac{\partial g}{\partial x}(1, x) = -2x$$

yields the following extreme values candidates

$$g(-1, 0) = -1, \quad g(1, 0) = -1, \quad g(0, -1) = 1, \quad g(0, 1) = 1$$

Finally, we look at the endpoints

$$g(-1, 1) = g(1, -1) = g(1, 1) = g(-1, -1) = 0$$

The function $g(x, y)$ has a maximum value 1 and a minimum value -1.

3.9 Extreme Values of Functions of n Variables

The process of finding extreme values can be generalised for arbitrary n. As in the two variable case, extreme values are assumed for continuous functions

of n variables in closed bounded regions; extrema are to be found on either critical interior points or at the boundary. The method for investigating the boundary of a function of several variables is identical to the two variable case, and will therefore not receive further attention.

In order to illustrate the pattern, we will, before generalising, consider the three variable case: $f(x, y, z)$. At critical points $\frac{\partial f}{\partial x}(a, b, c) = \frac{\partial f}{\partial y}(a, b, c) = \frac{\partial f}{\partial z}(a, b, c) = 0$ or one or several of the partial derivatives fail to exist.

Let H denote the Hessian matrix of second partial derivatives

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{pmatrix}$$

and let $D_1 = \frac{\partial^2 f}{\partial x^2}$, $D_2 = \begin{vmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{vmatrix}$ and $D_3 = det(H)$. Then

i) If $D_1 \geq 0, D_2 \geq 0$ and $D_3 \geq 0$ at (a, b, c), the point is a local minimum.

ii) If $D_1 \leq 0, D_2 \geq 0$ and $D_3 \leq 0$ at (a, b, c), the point is a local maximum.

iii) In any other case where $D_3 \neq 0$, f has a saddle point at (a,b,c).

The test includes the situation where the one or several of the determinants (D_1, D_2, D_3) equal zero. As a consequence, the critical points are not necessarily isolated - the situation is not generic.

For the general n variable case: $f(x_1, x_2, .., x_n)$. As previously, the critical points will be located where $\nabla f(\mathbf{c}) = \mathbf{0}$ or one or several of the partial derivatives fail to exist. Let H denote the Hessian matrix

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

and for each $k = 1, 2, ..., n$, let D_k denote the determinant of the corresponding Hessian matrix. Assume that $det(H) \neq 0$. Then [22]

i) If $D_k(\boldsymbol{c}) \geq 0 \quad \forall k$, the point is a local minimum.

ii) If $(-1)^k D_k(\boldsymbol{c}) \geq 0 \quad \forall k$, the point is a local maximum.

iii) Otherwise f has a saddle point at $\boldsymbol{c}$.

3.10 Mathematical Induction Principle

Mathematical induction is a proof technique commonly used to prove that a given proposition is true for for all positive integers n. Let $P_1, P_2, ..., P_n, ...$ be propositions, one for each positive integer, such that

1. P_1 is true.

2. for each positive integer n, P_n implies P_{n+1}.

Then P_n is true for each positive integer n. The first step is known as the base case, the second as the inductive step [24]

Chapter 4

The Hill Function

4.1 The Hill Function

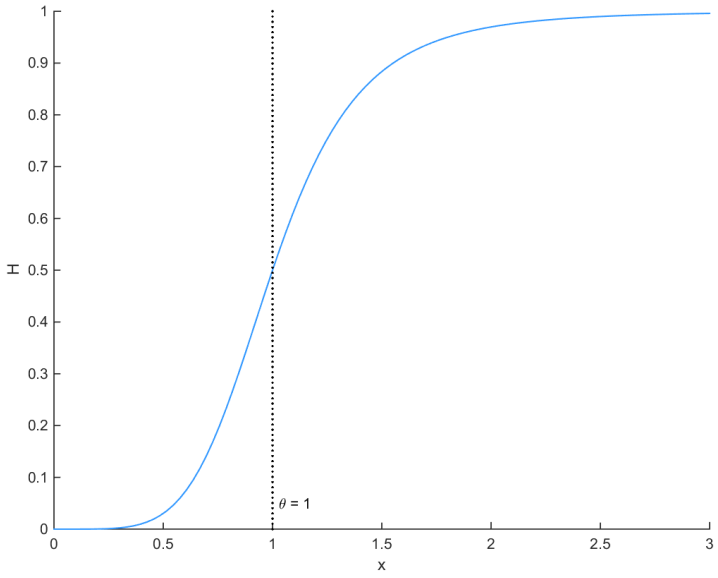


Figure 4.1: The Hill function (4.1) as a function of the gene concentration x, with threshold value $\theta = 1$ and steepness parameter $q = 0.2$.

The Hill function is given by

$$H(x, \theta, q) = \frac{x^{\frac{1}{q}}}{x^{\frac{1}{q}} + \theta^{\frac{1}{q}}}, \qquad x > 0, \;\; \theta > 0, \;\; q > 0 \tag{4.1}$$

where x is the concentration, θ is the threshold value and q is the steepness parameter. See figure 4.1.

Observe that

$$\mathrm{H(x,\ \theta,\ q)} \approx \begin{cases} 0, & \text{if } x < \theta \\ 1, & \text{if } x > \theta \end{cases}$$

Thus, $H(x, q, \theta) \in [0, 1]$.

4.2 The Metamodel

With a threshold value θ equal to 1, the Hill function (4.1) can be rewritten as

$$H(x, q) = \frac{x^{\frac{1}{q}}}{x^{\frac{1}{q}} + 1}, \qquad x > 0, \ \ q > 0 \tag{4.2}$$

where x is the concentration and q is the steepness parameter.

A metamodel of the Hill function can be developed based on function (4.2). The approximation will be a linear combination on the same form as (3.3)

$$\hat{H} = \sum_{i=1}^{k} t_i(q) p_i(x)^{\tau} \tag{4.3}$$

where t_i are the scores, p_i are the loadings, q is the steepness parameter, x is the concentration and k is the number of principal components.

Let H be a $I \times J$ matrix. With the discrete intervals $x_j = x_0, x_1, ..., x_J$ and $q_i = q_0, q_1, ..., q_I$, each element of H represents a function value of the function (4.2)

$$h_{ij} = \frac{x_j^{1/q_i}}{x_j^{1/q_i} + 1}$$

We then obtain the matrix

$$H_{I\times J} = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1J} \\ h_{21} & h_{22} & \cdots & h_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ h_{I1} & h_{I2} & \cdots & h_{IJ} \end{pmatrix}$$

We create the matrix H by working with the following fixed intervals for the concentration and the steepness parameter: $x_j \in [0.5, 1.5]$ and $q_i \in [0.1, 0.5]$. To determine the number of principal components, the error limit e_k is set to be 0.01. Or equivalently, $\sigma_{k+1} < 0.01$.

We ensure that the system is well-conditioned by studying the condition number κ_2. Notice that the condition number is forced to be high because $\sigma_{min} \leq 0.01$. However, the relative condition numbers between the components

$$\frac{\sigma_1}{\sigma_2}, \quad \frac{\sigma_2}{\sigma_3}, \quad \ldots, \quad \frac{\sigma_k}{\sigma_{k+1}}$$

will be small if the inverse condition number κ_2^{-1} is sufficiently low. The number of principal components also depends on the upper and lower limits of the concentration x and the steepness parameter q, and the corresponding step lengths.

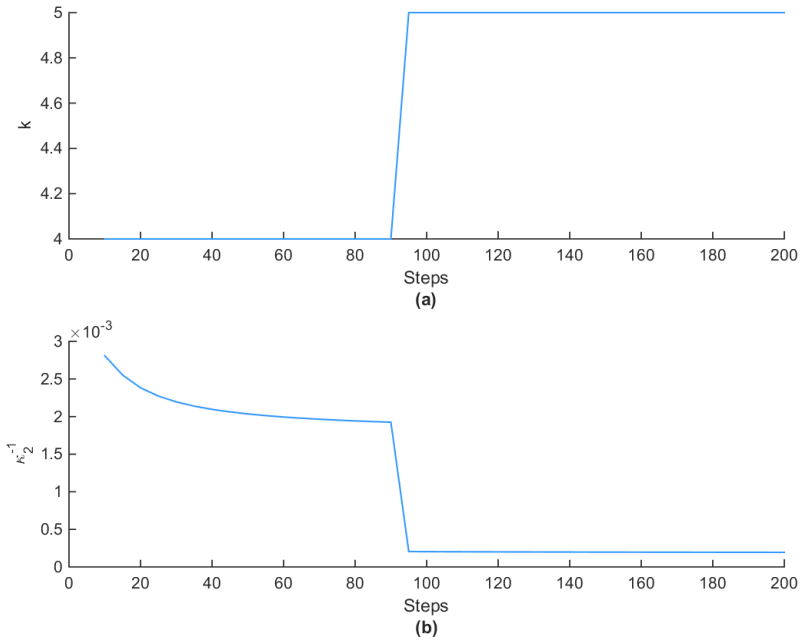


Figure 4.2: **(a)** The number of principal components k as a function of the number of steps $I = J$. **(b)** The inverse condition number κ_2^{-1} as a function of the number of steps $I = J$.

Figure 4.2 illustrates how the number of required principal components stabilizes at 5 when the number of steps reaches 100. This also applies to the

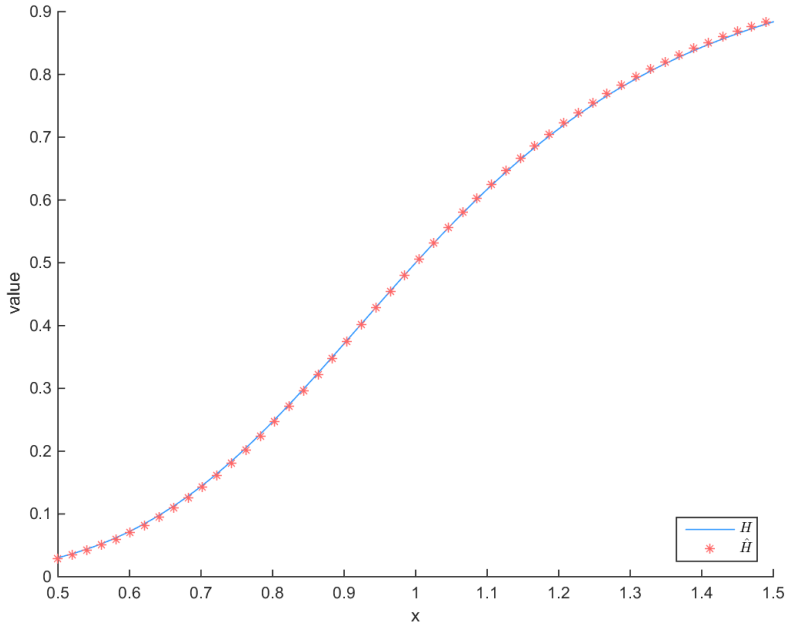


Figure 4.3: The Hill function (4.2) and the approximation of the Hill function (4.3) with 5 principal components and steepness parameter $q = 0.2$.

inverse condition number. When $I = J > 100$ the associated step lengths are $\Delta x < 0.01$ and $\Delta q < 0.004$. With this number of principal components, the original and approximated Hill function are illustrated in figure 4.3. The relative squared error, calculated by equation (3.8), is $\zeta^2 = 0.0086\%$.

Similarly, it is possible to study how the model behaves as the upper limits, x_J and q_I, increase and the lower limits, x_0 and q_0, decrease. When investigating the upper limits, it is convenient to keep the lower limits unchanged and vice versa. The upper limits pose no problems, see figure 4.4. Neither does the lower limit of x. A decreasing q_0 on the other hand, leads to a growing number of principal components. This is not unexpected. When $q \to 0$, the Hill function approaches the Heaviside step function; a function that is discontinuous at the threshold.

The inverse condition number remains in general small. We observe in figure 4.4 that the required number of principal components is unaffected by the lower limit of x. From now on we will therefore work with $x_j \in [0.01, 1.5]$. In order to preserve the step length $\Delta x < 0.01$, the number of division points is increased to $J = 150$. H is then a 100×150 matrix.

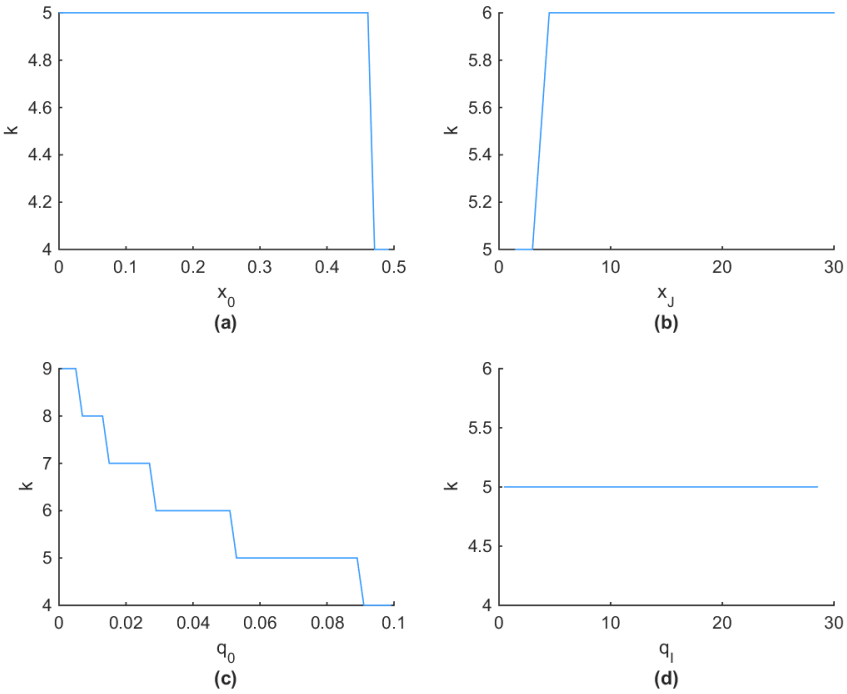


Figure 4.4: The number of principal components k as a function of the limits **(a)** x_0**(b)** x_J **(c)** q_0 and **(d)** q_J.

Chapter 5

The Scalar Case

5.1 A Genetic Network of One Gene

A genetic network of one gene can be modelled by an ordinary differential equation

$$\dot{x} = \underbrace{(az(x) + b)}_{\text{production term}} - \underbrace{(cz(x) + d)x}_{\text{degradation term}} \tag{5.1}$$

$$a + b \geq 0, \quad b \geq 0$$

where x is the concentration, a, b, c, d are coefficients, $z(x) = z(x, q)$ is the Hill function (4.2) and q is the steepness parameter.

The production term is assumed non negative. The term is linear, thus, the extrema are located at the endpoints. As the Hill function $z(x) \in [0, 1]$, it follows that also b is bound to be positive. The coefficient a determines the autoregulation. It is a positive feedback process when $a > 0$ and a negative when $a < 0$.

5.2 The Metamodel

For simplicity, we assume the coefficient $c = 0$. By combining the metamodel of the Hill function (4.3) with equation (5.1), the approximated differential equation for the concentration is

$$\dot{x} = a \sum_{i=1}^{k} t_i(q) p_i(x)^\tau + b - dx \tag{5.2}$$

where a, b, d are coefficients, t_i are the scores, p_i are the loadings, q is the steepness parameter and k is the number of principal components.

To compare the model (5.1) with the metamodel (5.2), we solve the differential equations. For this, Matlab has a function: *ode45*. This function can be directly applied for equation (5.1). However, *ode45* does not accept discrete functions. We therefore need to fit the metamodel of the Hill function by interpolation. This is done by the Matlab built in *interp1*. When interpolated, *ode45* can be applied also on equation (5.2), and the models analysed. This procedure is equivalent for systems of higher order.

Figure 5.1 illustrates how the values of a affect the solutions of the differential equations (5.1) and (5.2).

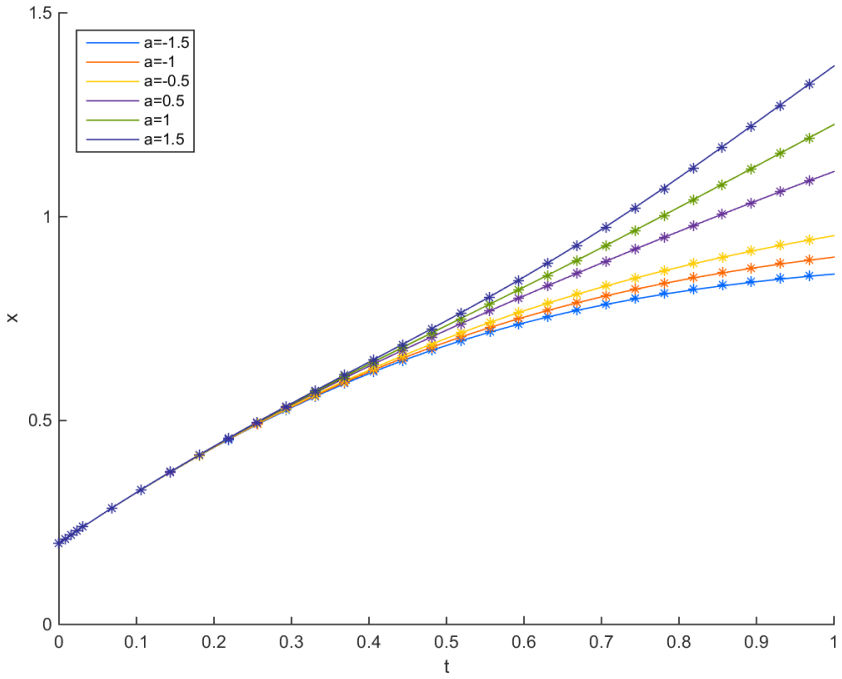


Figure 5.1: Solutions of the ordinary differential equations (5.1) (solid line) and (5.2) (asterisk) with the coefficients $b = 1.5$, $d = 1$, initial value $x_0 = 0.2$, steepness parameter $q = 0.2$ and different values for the coefficient a.

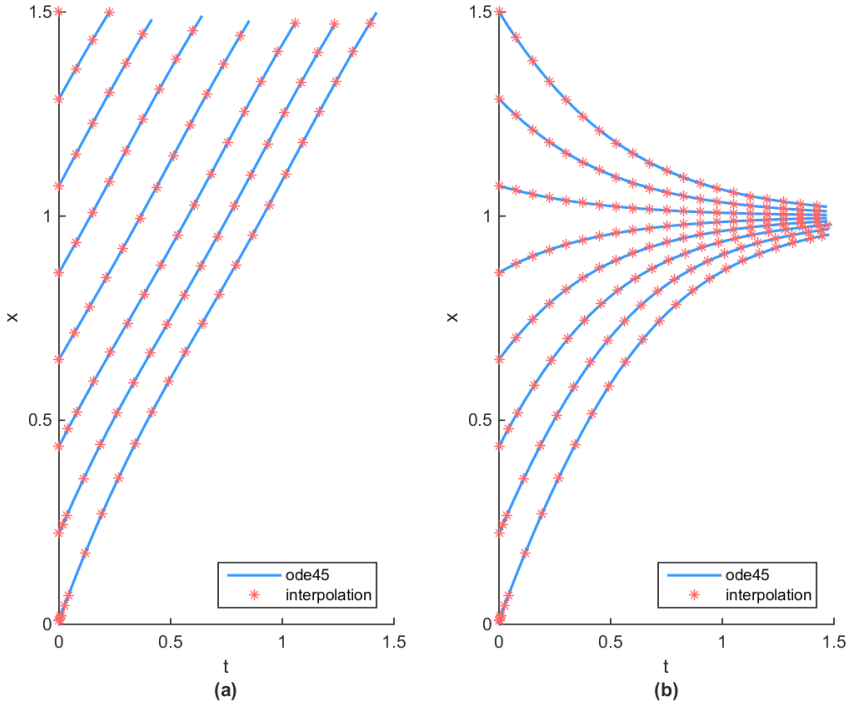


Figure 5.2: Solutions of the ordinary differential equations (5.1) and (5.2) with the coefficients $b = 1.5$, $d = 1$, steepness parameter $q = 0.2$ and initial conditions ranging from $x_0 = 0.01$ to $x_0 = 1.5$ with **(a)** $a = 1$ **(b)** $a = -1$

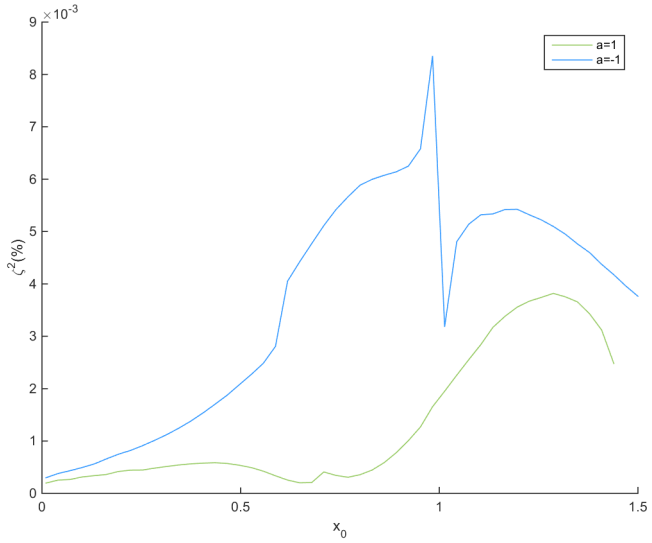


Figure 5.3: A plot of the relative squared errors ζ^2, equation (3.8), corresponding to the solutions illustrated in figure 5.2

The solutions for different initial values of x_0 are illustrated in figure 5.2. Notice that there is no solution for x when $x > 1.5$. This is due to the validity of the Hill function: the concentration x has an upper limit of 1.5 (described in section 4.2).

Attention should be directed towards the figures 5.1 and 5.2b; the solution x tends to have an asymptotic convergence to the threshold value $\theta = 1$ when the coefficient a is negative. The system appears to be in sliding mode. In this case though, it poses no problem.

Further, a more methodical investigation is of interest. The parameters are initially set to be $a = 1$, $b = 3$, $d = 1$, $q = 0.5$ and $x_0 = 0.2$, then they are changed one by one. See figure 5.4. Notice that the error does not exceed 0.25% under any of these circumstances. It seems reasonable to conclude that the model is a good approximation for the scalar case.

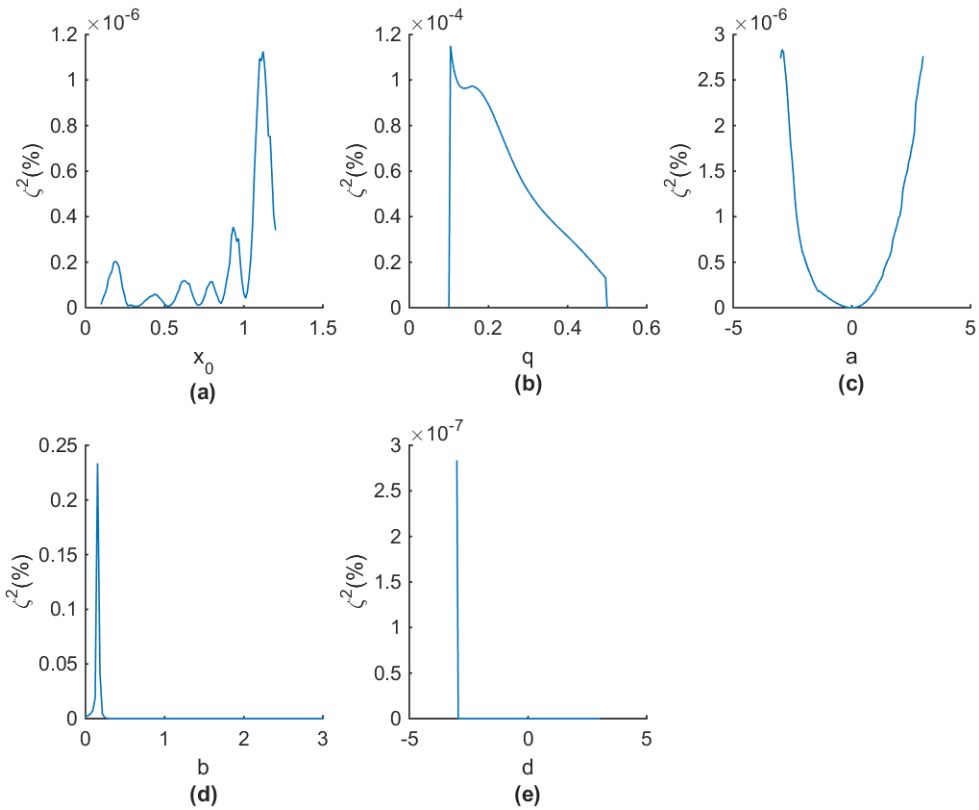


Figure 5.4: A plot of the relative squared error ζ^2, equation (3.8), for different **(a)** x_0 **(b)** q **(c)** a **(d)** b **(e)** d

Chapter 6

2×2 System of Genes

6.1 A Genetic Network of Two Genes

A genetic network of two genes can be described with the ordinary differential equations

$$\begin{aligned} \dot{x_1} &= (a_1 z_1 z_2 + b_1 z_1 + c_1 z_2 + d_1) - \gamma_1 x_1 \\ \dot{x_2} &= (a_2 z_1 z_2 + b_2 z_1 + c_2 z_2 + d_2) - \gamma_2 x_2 \\ a_1 z_1 z_2 + b_1 z_1 + c_1 z_2 + d_1 &\geq 0, \quad \gamma_1 \geq 0 \\ a_2 z_1 z_2 + b_2 z_1 + c_2 z_2 + d_2 &\geq 0, \quad \gamma_2 \geq 0 \end{aligned} \tag{6.1}$$

where x is the concentration, a, b, c, d, γ are coefficients and $z = z(x) = z(x, q)$ is the Hill function (4.2).

As in the scalar case (5.1), the first term is the production term and the second the degradation term. The production term is positive, and the coefficients a, b, c, d have to be chosen thereafter.

6.2 Analytic Determination of the Production Term

The production term of equation (6.1) is by definition positive. To ensure that this requirement is met, the coefficient d has to be set depending on the

other coefficients a, b and c. Thus, the problems is reduced to locating the minimum value of

$$f(z_1, z_2) = az_1z_2 + bz_1 + cz_2, \quad z_1, z_2 \in [0, 1]$$

There is only one critical interior point $(-\frac{c}{a}, -\frac{b}{a})$. This is a saddle point, as the second order derivatives are

$$\frac{\partial^2 f}{\partial z_1^2} = \frac{\partial^2 f}{\partial z_2^2} = 0, \quad \frac{\partial^2 f}{\partial z_1 \partial z_2} = a$$

No possible minimum values are found on the boundaries, whereas investigating the endpoints yields the following possible extreme values

$$f(1,0) = b, \quad f(0,1) = c, \quad f(0,0) = 0, \quad f(1,1) = a + b + c$$

Observe that all the minimum value candidates are found in the corners. Then d should be chosen so that $d \geq \mid min\{b, c, 0, a + b + c\} \mid$. A simple test to verify that this demand is fulfilled, is performed before the construction of the model.

6.3 The Metamodel

We combine the metamodel of the Hill function (4.3) with equation (6.1), and obtain the approximated differential equations

$$\begin{aligned} \dot{x_1} &= (a_1\hat{z_1}\hat{z_2} + b_1\hat{z_1} + c_1\hat{z_2} + d_1) - \gamma_1 x_1 \\ \dot{x_2} &= (a_2\hat{z_1}\hat{z_2} + b_2\hat{z_1} + c_2\hat{z_2} + d_2) - \gamma_2 x_2 \\ \hat{z_1} &= \sum_{i=1}^{k} t_i(q_1)p_i(x_1)^\tau, \quad \hat{z_2} = \sum_{i=1}^{k} t_i(q_2)p_i(x_2)^\tau \end{aligned} \tag{6.2}$$

where a, b, c, d, γ are coefficients, $\hat{z}$ is the approximated Hill function, t_i are the scores, p_i are the loadings, q is the steepness parameter and k is the number of principal components.

The approach is the same as for the scalar case - we compare the result from *ode45* with the interpolated solution of the approximation. From the tensor

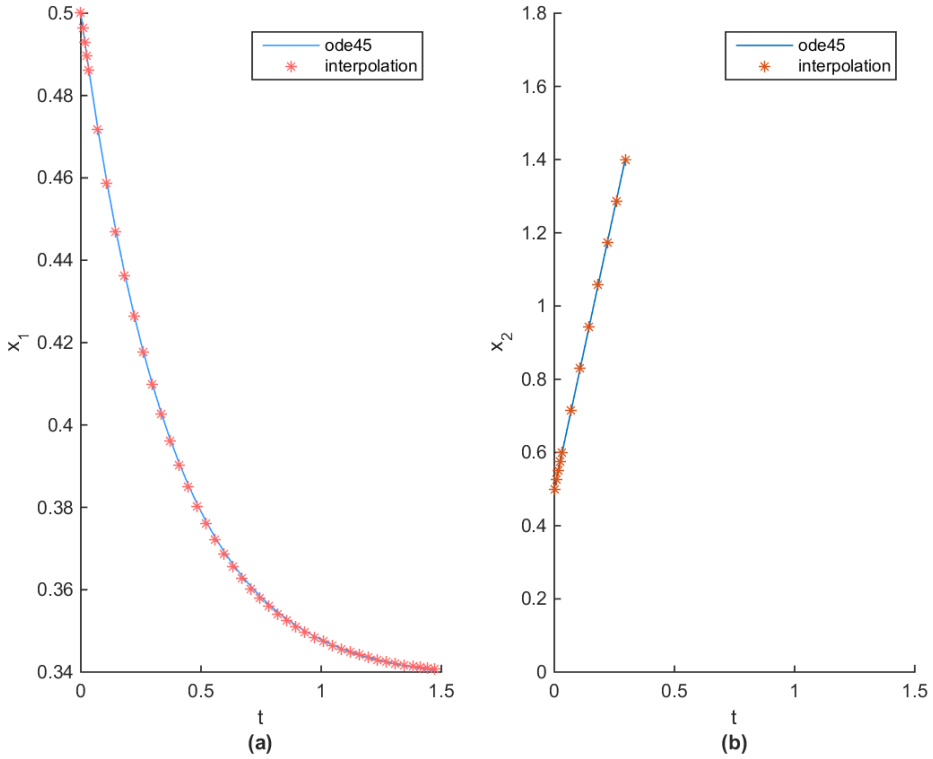


Figure 6.1: Solution of the ordinary differential equations (6.1) and (6.2) with the coefficients $a_1 = 1$, $a_2 = 2$, $b_1 = b_2 = 2$, $c_1 = 0$, $c_2 = 1$, $d_1 = 1$, $d_2 = 3$, $\gamma_1 = 3$, $\gamma_2 = 1$ initial value $x_0 = 0.5$ and steepness parameters $q_1 = 0.2, q_2 = 0.4$. **(a)** x_1 **(b)** x_2.

product we know that the product $\hat{z}_1\hat{z}_2$ requires no additional consideration. Figure 6.1 illustrates one solution of the ordinary differential equations (6.1) and (6.2). The relative squared errors are calculated using equation(3.8):

$$\zeta^2_{x_1} = 0.0060\% \quad and \qquad \zeta^2_{x_2} = 2.0 \times 10^{-4}\%$$

For various initial values x_0, the solutions of the systems (6.1) and (6.2) are illustrated in figure 6.2. In the scalar case we experienced a system in sliding mode. In figure 6.2b the same pattern can be observed; the system slides along the threshold value. This is further illustrated in the phase plane in figure 6.3.

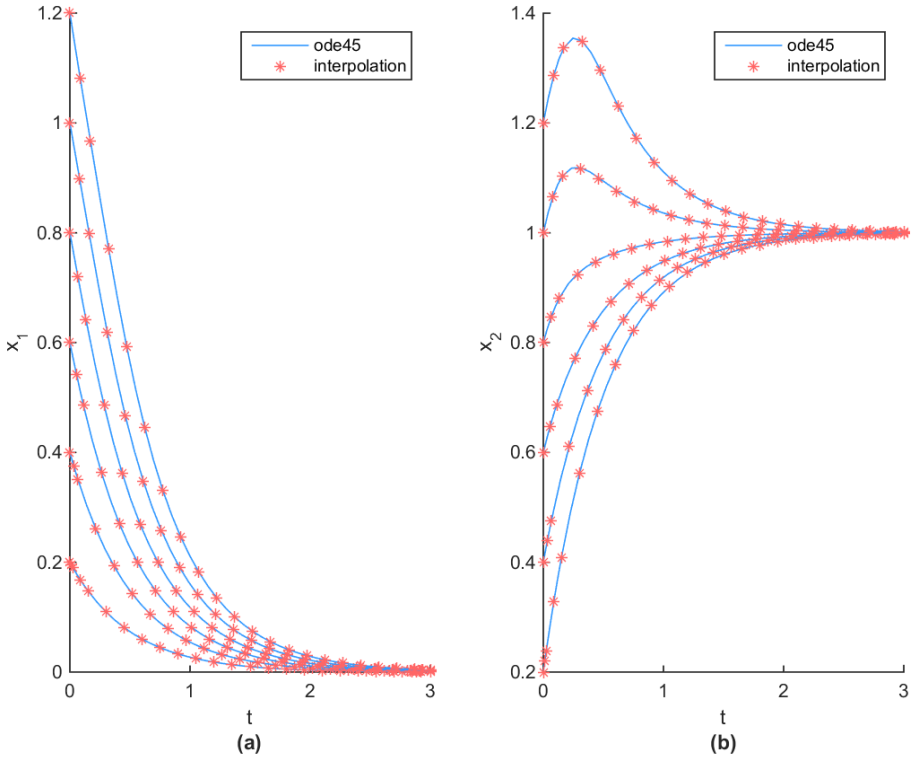


Figure 6.2: Solutions of the ordinary differential equations (6.1) and (6.2) with the coefficients $a_1 = -1$, $a_2 = 2$, $b_1 = 2$, $b_2 = 1$, $c_1 = 0$, $c_2 = -1$, $d_1 = 0$, $d_2 = 2$, $\gamma_1 = 2$, $\gamma_2 = 1$, steepness parameters $q_1 = 0.2$, $q_2 = 0.4$ and initial conditions ranging from $x_0 = 0.20$ to $x_0 = 1.20$. **(a)** x_1 **(b)** x_2.

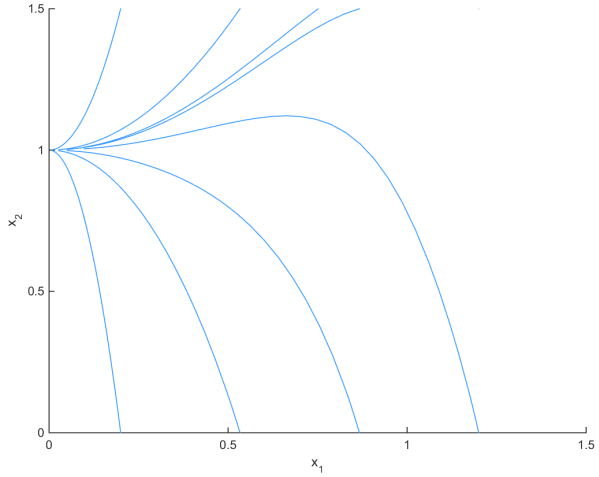


Figure 6.3: A plot of the solutions from figure 6.2 in the phase plane.

When studying the error more closely, it is convenient to focus on the terms of the highest order - these are the most dominant. Therefore, in equation (6.1), the coefficients b and c are now assumed zero. The other coefficients are chosen to be $a_1 = 1$, $a_2 = -1$, $d_1 = 3$, $d_2 = 5$ and $\gamma_1 = \gamma_2 = 5$. The parameters are examined individually. See figure 6.4. In all cases the relative squared error is within acceptable levels; the metamodel appears accurate.

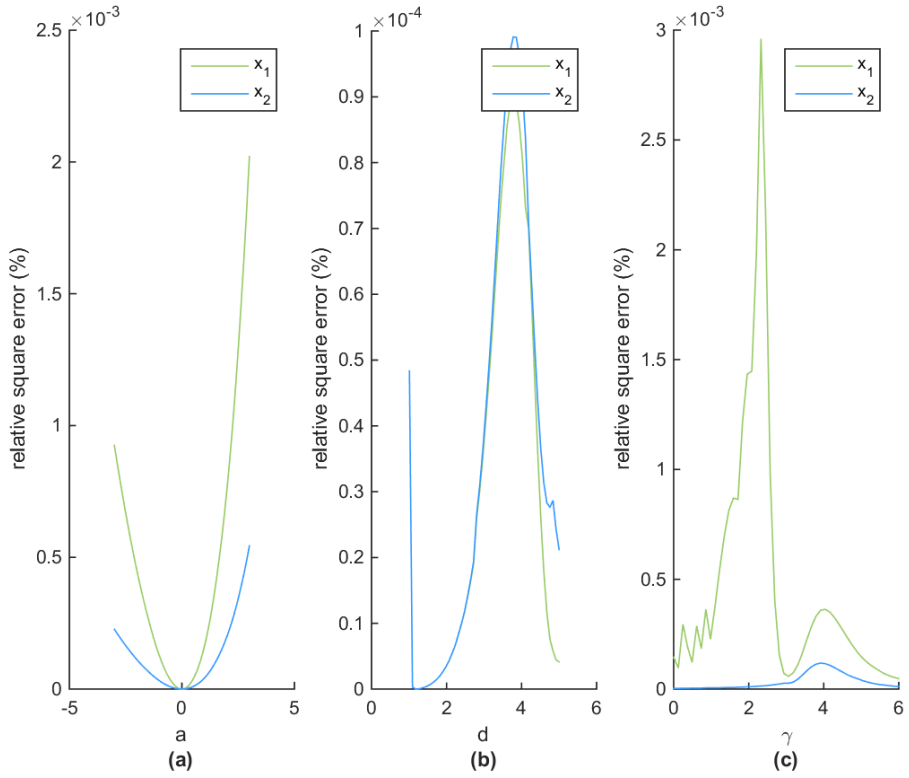


Figure 6.4: A plot of the relative squared error ζ^2, equation (3.8), for different **(a)** a **(b)** d **(c)** γ

6.4 A Benchmark Problem

The system from section 2.3 is actually a benchmark problem from the article "Analysis and generic properties of gene regulatory networks with graded response functions" by Plahte and Kjøglum [16].

This benchmark problem allows further evaluation and testing of the

metamodel. The benchmark model is

$$\begin{aligned} \dot{x_1} &= Z_1 + Z_2 - 2Z_1Z_2 - \gamma_1 x_1 \\ \dot{x_2} &= 1 - Z_1Z_2 - \gamma_2 x_2 \end{aligned} \tag{6.3}$$

where x is the concentration, γ_1 and γ_2 are positive parameters and Z_1 and Z_2 are the Hill function (4.2) with steepness parameter $q = 0.08$. Notice that

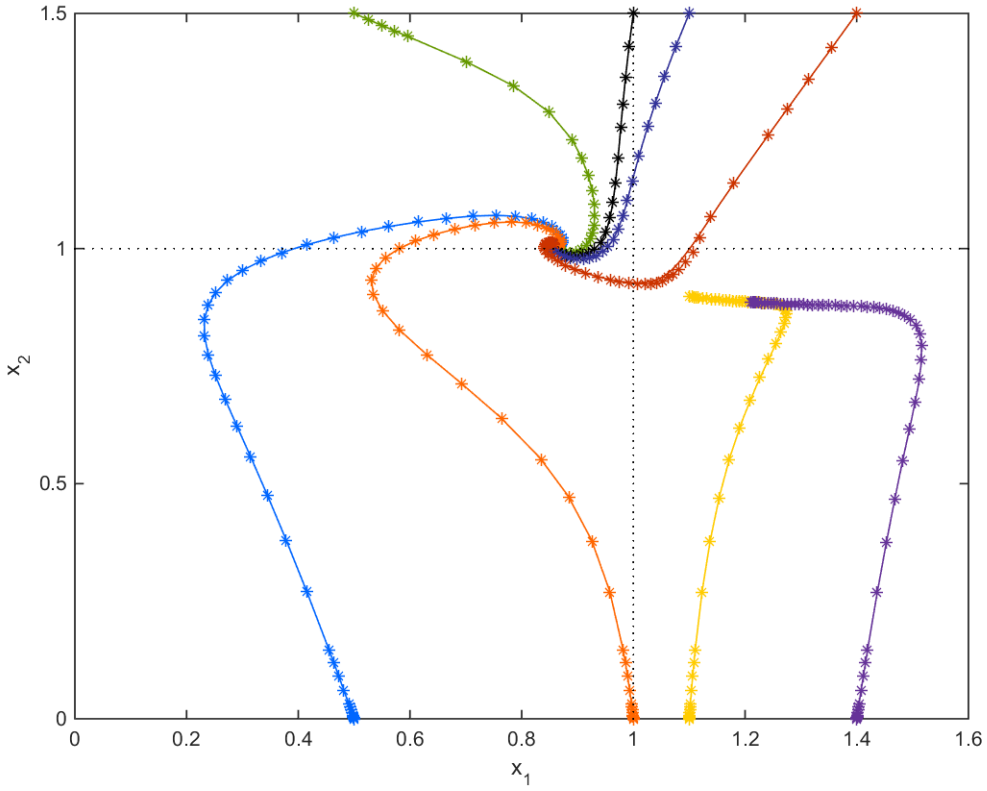


Figure 6.5: Trajectories in the phase plane corresponding to the solutions of the system (6.3). The solid line represents the original trajectories [16]. The asterisk represents the metamodel.

the steepness parameter in the benchmark problem is $q = 0.08$, while the metamodel has a lower limit of $q = 0.1$. For this reason, the the steepness parameter is set to be $q = 0.1$ for both for the benchmark problem and the approximated metamodel. The result is illustrated in figure 6.5.

The analysis of the benchmark problem supports previous results; the metamodel behaves well also for the 2x2 system.

Chapter 7

Modelling Circadian Oscillations

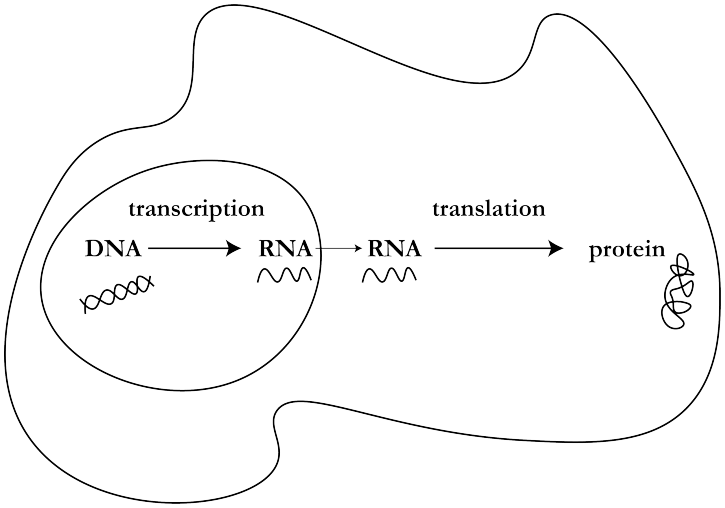


Figure 7.1: Simplified schematic model of the process from DNA to protein [3].

One of the most fundamental concepts of molecular biology, is the process from DNA to protein. A simple overview is illustrated in figure 7.1. In the nucleus of eukaryotic cells, mRNA is synthesized from one strand of a DNA helix. After the transcription of DNA to mRNA, the translation begins. The ribosome decodes the mRNA and, with help of the tRNA molecules transporting amino acids, produce an amino acid chain - a polypeptide. Finally the polypeptide folds into a protein [3, 15].

This process is essential in the article "A Model of Circadian Oscillations in the Drosophila Period Protein (PER)" by Albert Goldbeter [2]. A benchmark problem for a 5x5 system is presented and, as the title implies, a

theoretical model describing the mechanisms of circadian oscillations in the period protein (PER) in Drosophila (a genus of small flies) is discussed.

In order to keep the model simple, only three states of the period protein are considered: un- (P_0), mono- (P_1) and bisphosphorylated (P_2) [2]. The system also depends on the concentration of PER mRNA, M, and PER, P_N, in the nucleus. Figure 7.2 illustrates the benchmark model for circadian oscillations in PER and PER mRNA. The system is described by the ordinary differential equations

$$\begin{aligned}
\dot{M} &= v_s \frac{K_I^n}{K_I^n + P_N^n} - v_m \frac{M}{K_m + M} \\
\dot{P}_0 &= k_s M - V_1 \frac{P_0}{K_1 + P_0} + V_2 \frac{P_1}{K_2 + P_1} \\
\dot{P}_1 &= V_1 \frac{P_0}{K_1 + P_0} - V_2 \frac{P_1}{K_2 + P_1} - V_3 \frac{P_1}{K_3 + P_1} + V_4 \frac{P_2}{K_4 + P_2} \\
\dot{P}_2 &= V_3 \frac{P_1}{K_3 + P_1} - V_4 \frac{P_2}{K_4 + P_2} - k_1 P_2 + k_2 P_N - v_d \frac{P_2}{K_d + P_2} \\
\dot{P}_N &= k_1 P_2 - k_2 P_N
\end{aligned} \tag{7.1}$$

where M is the concentration of PER mRNA, P_0, P_1 and P_2 are the concentrations of phosphorylated PER and P_N is the concentration of PER in the nucleus. The parameter values are: $n = 4$, $v_s = 0.76\mu M h^{-1}$, $v_m = 0.65\mu M h^{-1}$, $v_d = 0.95\mu M h^{-1}$, $k_s = 0.38 h^{-1}$, $k_1 = 1.9 h^{-1}$, $k_2 = 1.3 h^{-1}$, $K_1 = K_2 = K_3 = K_4 = 2\mu M h^{-1}$, $K_I = 1\mu M h^{-1}$, $K_d = 0.2\mu M h^{-1}$, $V_1 = 3.2\mu M h^{-1}$, $V_2 = 1.58\mu M h^{-1}$, $V_3 = 5\mu M h^{-1}$ and $V_4 = 2.5\mu M h^{-1}$.

The negative feedback exerted by the PER in the nucleus, P_N, on the concentration of PER mRNA, M, is fundamental to the mechanism of oscillations. In equation (7.1) this is represented by the first term in the differential equation $\dot{M}$.

Notice that the negative feedback term is not discretized in the meta-model, but represented by the given continuous function. The other terms are replaced by the approximated Hill function. However, some modifications must be applied.

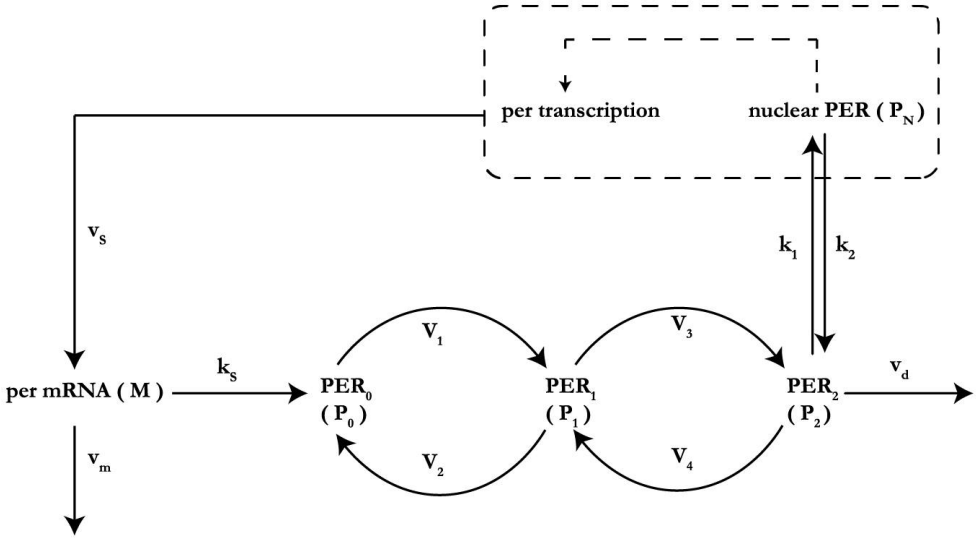


Figure 7.2: Scheme of the model for circadian oscillations in PER and PER mRNA [2] corresponding to equation (7.1). The dashed line represents the nucleus.

In equation (7.1) sigmoids are on the form

$$S(P, K) = \frac{P}{K + P} \tag{7.2}$$

where $P \in \{M, P_0, P_1, P_2\}$ and $K \in \{K_m, K_1, K_2, K_3, K_4, K_d\}$.

The metamodel of the Hill function is based on equation (4.2); a special case of equation (7.2) with $K = 1$. Therefore, in order to represent equation (7.2) by the approximation of the Hill function, we substitute x by $\frac{P}{K}$ and set the steepness parameter q equal to one. We obtain

$$H(x, 1) = \frac{x}{x + 1} = S(P, K), \qquad x = \frac{P}{K} \tag{7.3}$$

where H is the Hill function, x is the steepness parameter, S is the sigmoid (7.2) and P and K are parameters.

From section 4.2 the given intervals for x and q are: $x_j \in [0.01, 1.5]$ and $q_i \in [0.1, 0.5]$. Extending these intervals is necessary in order to model the benchmark problem properly. Firstly, the model have to include $q = 1$, and secondly, the upper limit of x should be increased to at least $x_J = 3.0$. From

figure 4.4 we know that changing the upper limit q_I has no consequences, while increasing the upper limit x_J will require one more principal component.

By increasing the number of division points to $J = 299$ and $I = 225$, the step lengths $\Delta x < 0.01$ and $\Delta q < 0.004$ are preserved. The new intervals are: $x_j \in [0.01, 3.0]$ and $q_i \in [0.1, 1]$. H is then a 225×299 matrix and the metamodel has 6 principal components.

With the substitution (7.3) and the new intervals for x and q, the metamodel for the benchmark problem (7.1) can be constructed. The result is illustrated in figure (7.3). We observe oscillations, which is pleasing, but the metamodel clearly deviates from the original model. What we observe resemble the Gibbs phenomenon: deviation (typically irregular fluctuations) close to the limits of the model - let us call it a generalised Gibbs phenomenon.

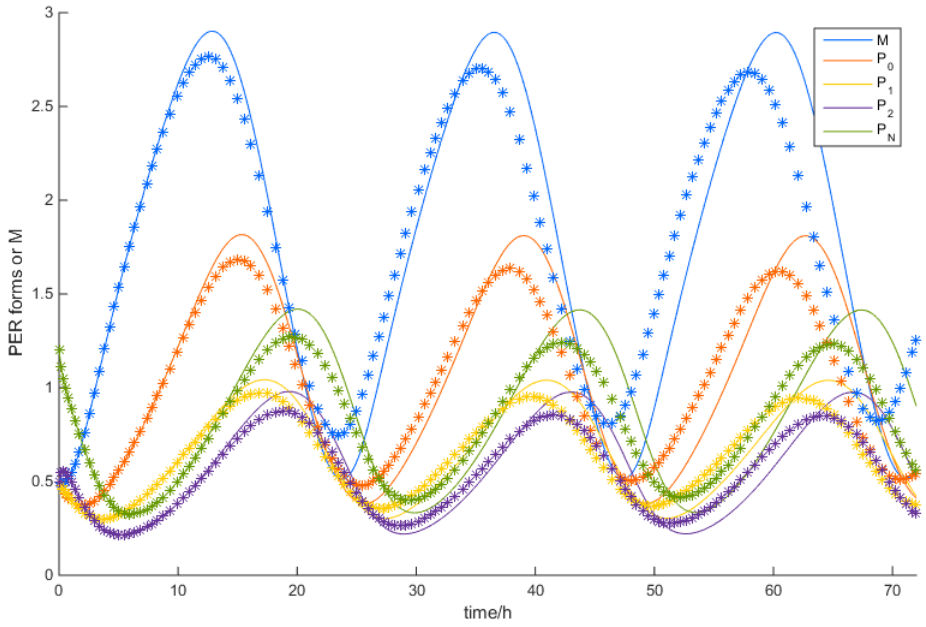


Figure 7.3: Solutions of the system (7.1). The solid lines represent the original oscillations [2]. The asterisk represent the metamodel where $x_j \in [0.01, 3.0]$. The concentration scale is given in μM.

Particularly the concentration of PER mRNA is near the upper limit $x_J = 3.0$. It therefore seems reasonable to increase the upper limit x_J further. With H as a 225×399 matrix, the interval for x is extended to $x_j \in [0.01, 4.0]$. It is easy to see that the approximation illustrated in figure 7.4 is more accurate than figure 7.3. This may be regarded satisfactory.

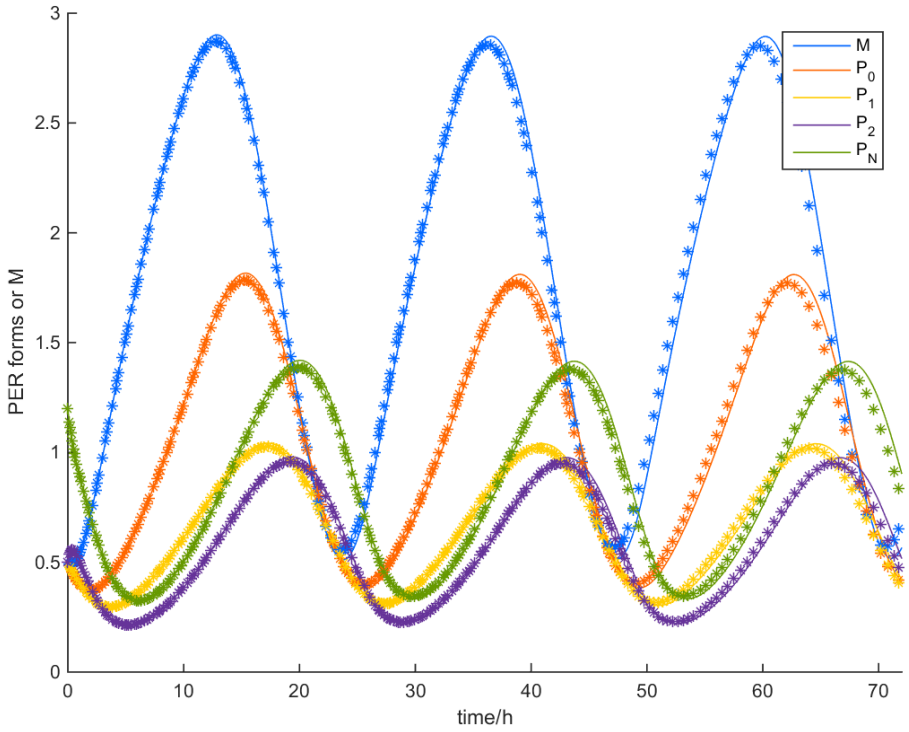


Figure 7.4: Solutions of the system (7.1). The solid lines represent the original oscillations [2]. The asterisk represent the metamodel where $x_j \in [0.01, 4.0]$. The concentration scale is given in μM.

Chapter 8

3×3 System of Genes

8.1 A Genetic Network of Three Genes

Let us revisit a genetic network of the general type we studied earlier. The following differential equations model a genetic network of three genes

$$\begin{aligned}
\dot{x_1} &= (a_1 z_1 z_2 z_3 + b_1 z_1 z_2 + c_1 z_1 z_3 + d_1 z_2 z_3 + \\
&\qquad e_1 z_1 + f_1 z_2 + g_1 z_3 + h_1) - \gamma_1 x_1 \\
\dot{x_2} &= (a_2 z_1 z_2 z_3 + b_2 z_1 z_2 + c_2 z_1 z_3 + d_2 z_2 z_3 + \\
&\qquad e_2 z_1 + f_2 z_2 + g_2 z_3 + h_2) - \gamma_2 x_2 \\
\dot{x_3} &= (a_3 z_1 z_2 z_3 + b_3 z_1 z_2 + c_3 z_1 z_3 + d_3 z_2 z_3 + \\
&\qquad e_3 z_1 + f_3 z_2 + g_3 z_3 + h_3) - \gamma_3 x_3
\end{aligned} \tag{8.1}$$

$$a_j z_1 z_2 z_3 + b_j z_1 z_2 + c_j z_1 z_3 + d_j z_2 z_3 + e_j z_1 + f_j z_2 + g_j z_3 + h_j \geq 0,$$

$$\gamma_j \geq 0, \quad j = 1, 2, 3$$

where x is the concentration, $a, b, c, d, e, f, g, h, \gamma$ are coefficients and $z = z(x) = z(x, q)$ is the Hill function (4.2).

The first term is the positive production term and the second is the degradation term.

8.2 The Metamodel

The metamodel of the 3x3 system is constructed the same way as previously with the 2x2 system; from the metamodel of the Hill function (4.3) and equation (8.1), the following approximated differential equations are yielded

$$\begin{aligned}
\dot{x_1} &= (a_1\hat{z_1}\hat{z_2}\hat{z_3} + b_1\hat{z_1}\hat{z_2} + c_1\hat{z_1}\hat{z_3} + d_1\hat{z_2}\hat{z_3} + \\
&\quad e_1\hat{z_1} + f_1\hat{z_2} + g_1\hat{z_3} + h_1) - \gamma_1 x_1 \\
\dot{x_2} &= (a_2\hat{z_1}\hat{z_2}\hat{z_3} + b_2\hat{z_1}\hat{z_2} + c_2\hat{z_1}\hat{z_3} + d_2\hat{z_2}\hat{z_3} + \\
&\quad e_2\hat{z_1} + f_2\hat{z_2} + g_2\hat{z_3} + h_2) - \gamma_2 x_2 \\
\dot{x_3} &= (a_3\hat{z_1}\hat{z_2}\hat{z_3} + b_3\hat{z_1}\hat{z_2} + c_3\hat{z_1}\hat{z_3} + d_3\hat{z_2}\hat{z_3} + \\
&\quad e_3\hat{z_1} + f_3\hat{z_2} + g_3\hat{z_3} + h_3) - \gamma_3 x_3
\end{aligned} \tag{8.2}$$

$$\hat{z_1} = \sum_{i=1}^{k} t_i(q_1)p_i(x_1)^\tau, \quad \hat{z_2} = \sum_{i=1}^{k} t_i(q_2)p_i(x_2)^\tau,$$
$$\hat{z_3} = \sum_{i=1}^{k} t_i(q_3)p_i(x_3)^\tau$$

where $a, b, c, d, e, f, g, h, \gamma$ are coefficients, $\hat{z}$ is the approximated Hill function, t_i are the scores, p_i are the loadings, q is the steepness parameter and k is the number of principal components.

Figure 8.1 illustrates a solution of the ordinary differential equations (8.1) and (8.2). The coefficients are chosen as listed in table 8.1. The relative squared errors are calculated using equation(3.8):

$$\zeta_{x_1}^2 = 0.80 \times 10^{-4}\%, \qquad \zeta_{x_2}^2 = 0.028\% \qquad and \qquad \zeta_{x_3}^2 = 0.0024\%$$

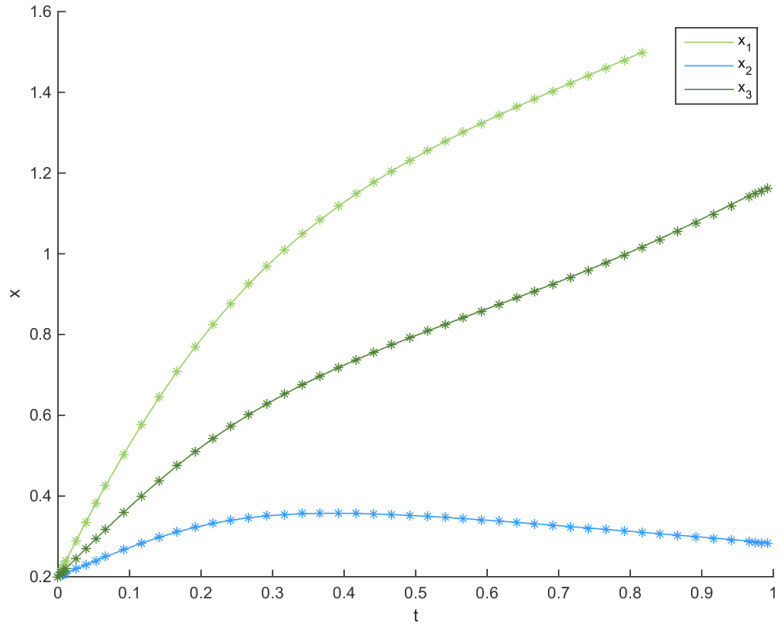


Figure 8.1: Solution of the ordinary differential equations (8.1) (solid line) and (8.2) (asterisk) with initial value $x_0 = 0.2$ and coefficients from table 8.1

Table 8.1: Table of coefficients corresponding to figure 8.1

	a_j	b_j	c_j	d_j	e_j	f_j	g_j	h_j	γ_j	q_j
$j = 1$	-2	1	0	2	-1	1	1	4	2	0.2
$j = 2$	2	1	-1	-2	-1	-1	1	1	1	0.4
$j = 3$	0	1	0	2	-1	1	1	2	1	0.1

For the purpose of studying the 3x3 model more closely, we consider a system where only the coefficients representing the terms of highest order are non zero. Thus, in equation (8.1), the coefficients b, c, d, e, f and g equals zero. The other coefficients are initially set to be $a_1 = 2, a_2 = -1, a_3 = -3, h_1 = 3, h_2 = 5, h_3 = 7$ and $\gamma_1 = \gamma_2 = \gamma_3 = 3$. The result is illustrated in figure 8.2. Attention must be directed to figure 8.2b. When $h \approx 5$, the solution x_3 of the metamodel has a relative squared error of almost 16%. Such behaviour requires further analysis.

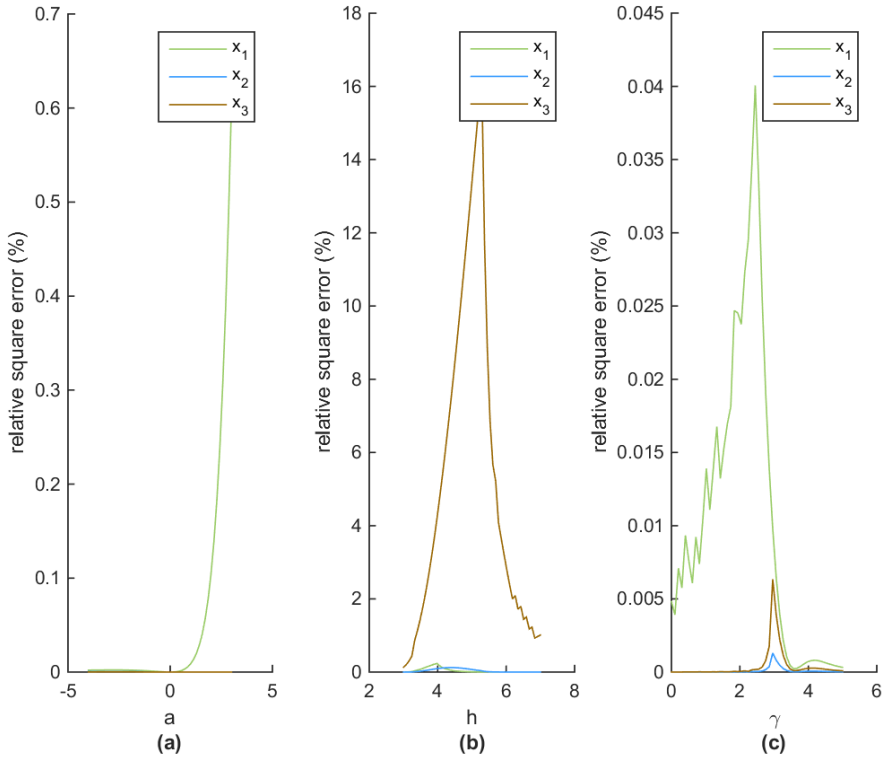


Figure 8.2: A plot of the relative squared error ζ^2, equation (3.8), for different **(a)** a **(b)** h **(c)** γ

8.3 Sources of Error in the Metamodel

Figure 8.3 illustrates a solution of the ordinary differential equations (8.1) and (8.2) where the relative squared error for the solution x_3 is above 20%. Apparently, there is a flaw in the metamodel for the 3x3 system.

Table 8.2: Table of coefficients corresponding to figure 8.3

	a_j	b_j	c_j	d_j	e_j	f_j	g_j	h_j	γ_j	q_j
$j=1$	1	0	0	0	0	0	0	5	1	0.2
$j=2$	-1	0	0	0	0	0	0	5	1	0.4
$j=3$	-4	0	0	0	0	0	0	5	1	0.1

The discretization of the function and the interpolation are both possible sources of error. With the coefficients from table 8.2, the differential

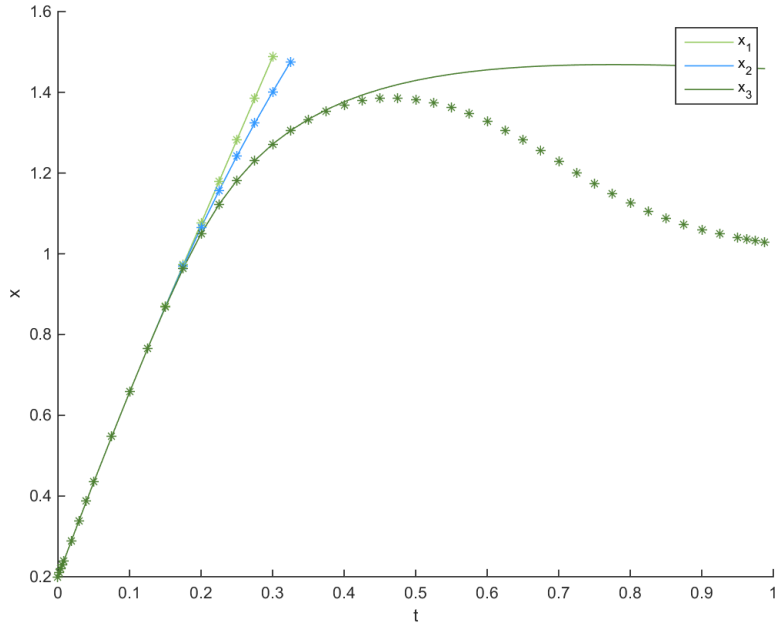


Figure 8.3: Solution of the ordinary differential equations (8.1) (solid line) and (8.2) (asterisk) with initial value $x_0 = 0.2$ and coefficients from table 8.2

equations for x_3 are

$$\begin{aligned} \dot{x_3} &= (-4z_1z_2z_3 + 5) - x_3 \\ \dot{x_3} &= (-4\hat{z_1}\hat{z_2}\hat{z_3} + 5) - x_3 \end{aligned} \tag{8.3}$$

where x is the concentration, $z = z(x) = z(x, q)$ is the Hill function (4.2) and $\hat{z}$ is the approximated Hill function.

A comparison of the right side of equation (8.3), the discretization and the interpolation of the discretization is illustrated in figure 8.4. If this was the source of error, deviation should be observed.

This requires further exploration; we perform a comparison between the step vise derivative of the solution and the derivative yielded by insertion.

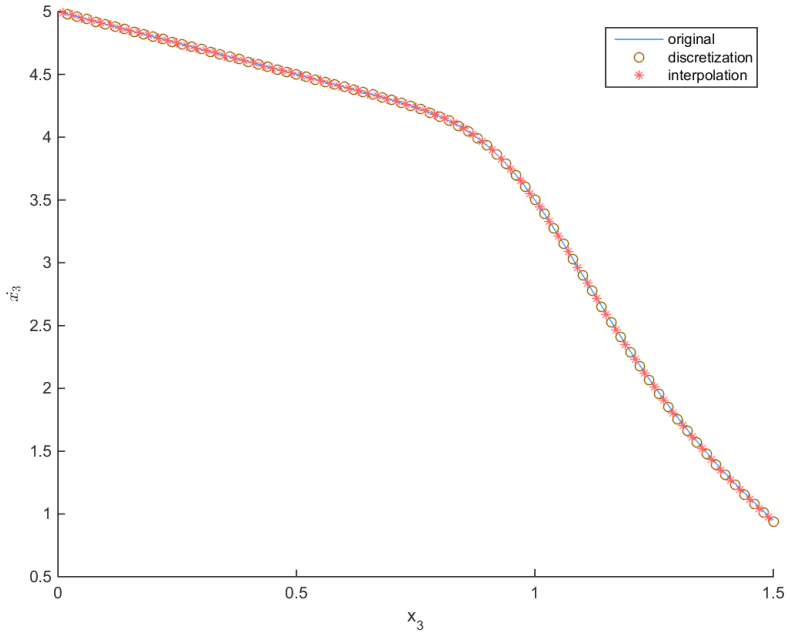


Figure 8.4: Plot of the right side of equation (8.3): the original model, the discretization and the interpolation of the discretization.

.

The step vise derivative $\hat{\dot{x}}_k$ is approximated by the formula

$$\hat{\dot{x}}_k \approx \frac{\hat{x}_k - \hat{x}_{k-1}}{h}, \qquad \hat{\mathbf{x}} = \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_N \end{pmatrix}, \qquad k = 2, 3, ..., N \tag{8.4}$$

where $\hat{x}$ is the solution vector, N is the length, $\hat{x}_k$ is the k-th element and h is the step length.

The solutions $\hat{x}_1$, $\hat{x}_2$ and $\hat{x}_3$ of the equations (8.2), with coefficients from table 8.2, are inserted into the equations to estimate the derivative $\dot{x}_{inserted}$

$$\begin{aligned} \dot{x}_{1inserted} &= (\hat{z}_1\hat{z}_2\hat{z}_3 + 5) - \hat{x}_1 \\ \dot{x}_{2inserted} &= (-\hat{z}_1\hat{z}_2\hat{z}_3 + 5) - \hat{x}_2 \\ \dot{x}_{3inserted} &= (-4\hat{z}_1\hat{z}_2\hat{z}_3 + 5) - \hat{x}_3 \end{aligned} \tag{8.5}$$

where $\hat{x}$ is the solution of equation (8.2) and $\hat{z}$ is the approximated Hill function.

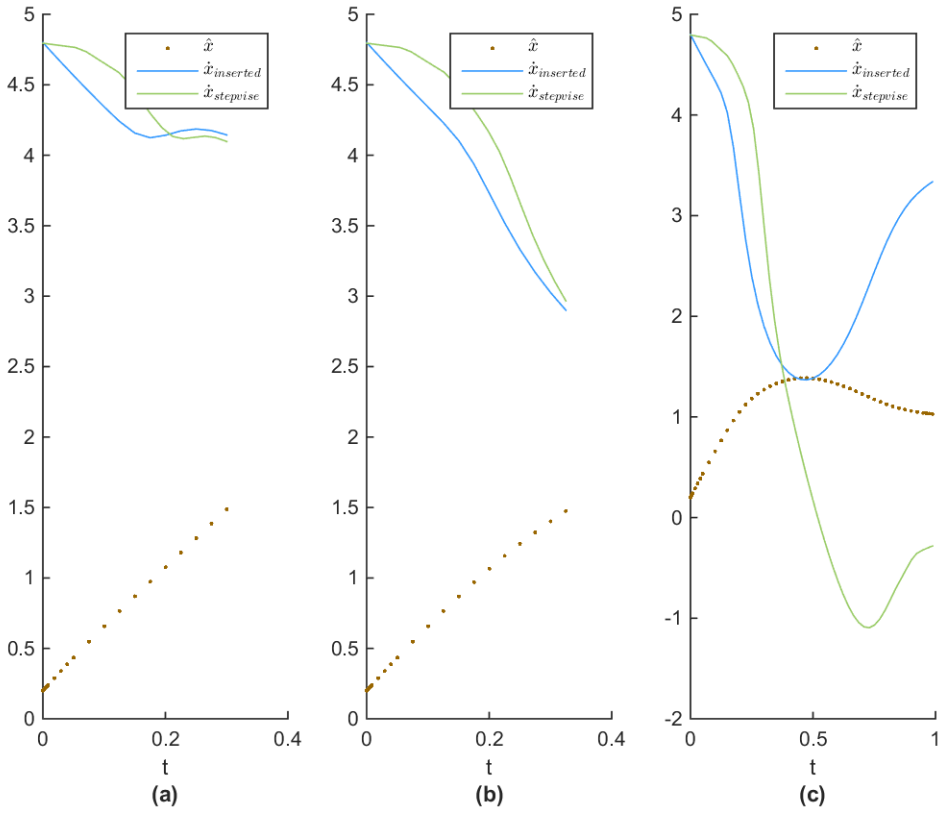


Figure 8.5: A plot of the solutions of equation (8.2), the derivative by insertion, equation (8.5), and the step vise derivative, equation (8.4), for **(a)** x_1 **(b)** x_2 **(c)** x_3. The initial value is $x_0 = 0.2$ and the coefficients are from table 8.2

The equations (8.5) can then be compared with the step vise derivative $\dot{x}_{stepvise}$ of $\hat{x}$, approximated by equation (8.4). This is illustrated in figure 8.5.

Figure 8.5c shows that there is an error in the derivative. While in figure 8.5a and 8.5b the derivatives $\dot{x}_{inserted}$ and $\dot{x}_{stepvise}$ are rather similar, they clearly deviate for x_3. With figure 8.3 in mind, this should not be surprising. The interpolation of the function appears to be correct. The derivatives, however, are not.

Previously, in chapter 7, we looked at a system of higher order. In that case, extending the metamodel was inevitable. Most importantly, the upper limit x_J was increased from $x_J = 1.5$ to $x_J = 4.0$. Notice that in figure 8.3, the solution x_3 of system (8.1) is very close to the upper limit of the metamodel. It is not unlikely that the deviation is, as in the modelling of circadian oscillations, a case of the generalised Gibbs phenomenon.

Let us in the equations (8.2) replace the approximation of the Hill function with the extended metamodel from the benchmark problem. With a relative squared error $\zeta^2_{x_3} = 0.029\%$, the result is pleasing. See figure 8.6. The source of error can be considered identified. Moreover, a weakness is revealed; the metamodel should not be used uncritically when close to the limits of x.

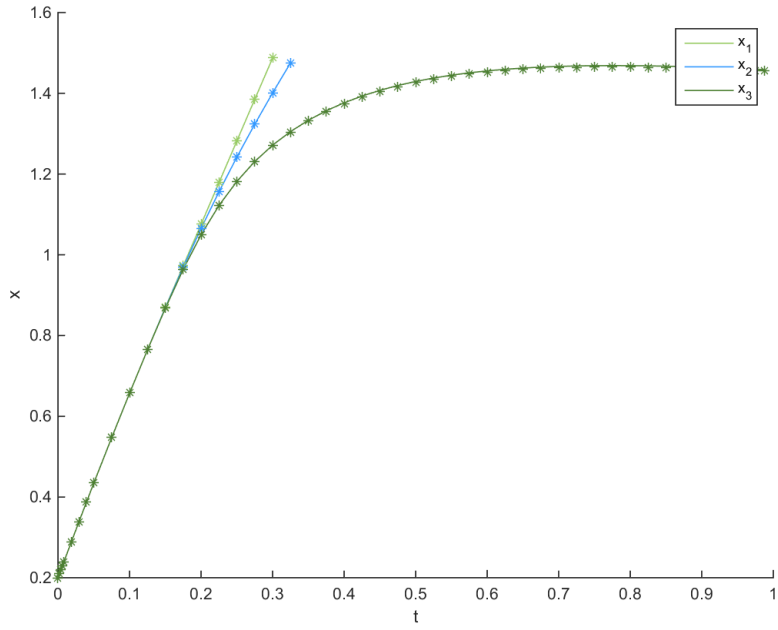


Figure 8.6: Solution of the ordinary differential equations (8.1) (solid line) and (8.2) (asterisk) with the extended metamodel. The initial value is $x_0 = 0.2$ and the coefficients are from table 8.2

Chapter 9

Determination of the Production Term for a $n \times n$ System of Genes

The production terms of the general genetic systems are positive. The constants in these terms are therefore determined by locating the minima of the non-constant part. In both the genetic network of one and two genes, the extreme values are obtained at the corner points. It is possible to prove that this also applies to a network of n genes. The production term for a $n \times n$ genetic network can be expressed

$$\begin{aligned} f_n(z_1, z_2, ..., z_n) = & \, a z_1 z_2 \cdots z_n + b_1 z_2 z_3 \cdots z_n \\ & + b_2 z_1 z_3 z_4 \cdots z_n + ... + b_{\binom{n}{n-1}} z_1 z_2 \cdots z_{n-1} \\ & + c_1 z_3 z_4 \cdots z_n + ... + c_{\binom{n}{n-2}} z_1 z_2 \cdots z_{n-2} \\ & + ... + p_1 z_1 z_2 + ... + p_{\binom{n}{2}} z_{n-1} z_n \\ & + q_1 z_1 + q_2 z_2 + ... + q_n z_n \end{aligned} \tag{9.1}$$

$$z_i \in [0, 1], \quad i = 1, 2, .., n$$

where $a, b, c, .., q$ are coefficients and z_i is the Hill function. Notice that some of the subscripts are denoted by the binomial coefficient [1]

The proposition is that the maximum and minimum values of f_n are obtained in the corner points. That is where all $z_i \in \{0, 1\}$. By mathematical induction, these two steps follows

[1] There are $\binom{N}{K} = \frac{N!}{K!(N-K)!}$ ways to pick an unordered set of K elements from a set of N elements [20].

1. Extrema are obtained in the corner points when $n = 1$.
2. Assume that extrema are obtained in the corner points for the n variable case. Show that the same holds for $n + 1$.

Firstly the base case. From equation (9.1) with $n = 1$ we get

$$f_1(z_1) = az_1, \qquad z_1 \in [0, 1]$$

The function is linear and locating the extreme values is trivial; the extrema are obtained in the endpoints when $n = 1$.

Secondly the inductive step. For $n+1$ equation (9.1) can be rewritten as

$$f_{n+1}(z_1, z_2, ..., z_{n+1}) = z_{n+1} f_n^1(z_1, ..., z_n) + f_n^2(z_1, ..., z_n)$$

with the corresponding $(n + 1) \times (n + 1)$ Hessian matrix

$$H = \begin{pmatrix} 0 & z_{n+1}\frac{\partial f_n^1}{\partial z_1 \partial z_2} + \frac{\partial f_n^2}{\partial z_1 \partial z_2} & \cdots & \cdots & \frac{\partial f_n^1}{\partial z_1} \\ z_{n+1}\frac{\partial f_n^1}{\partial z_2 \partial z_1} + \frac{\partial f_n^2}{\partial z_2 \partial z_1} & 0 & \cdots & \cdots & \frac{\partial f_n^1}{\partial z_2} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \cdots & 0 & \frac{\partial f_n^1}{\partial z_n} \\ \frac{\partial^2 f_n^1}{\partial z_1} & \frac{\partial f_n^1}{\partial z_1} & \cdots & \frac{\partial f_n^1}{\partial z_n} & 0 \end{pmatrix}$$

From the theory about extreme values of functions of n variables, we know how to classify critical interior points by use of determinants of the Hessian matrix. Also, a stable, genetic model is generic. For isolated extreme values, a local minimum has only positive determinants, while a local maximum has determinants alternating between positive and negative values:

$$\text{Local minimum: } D_1 > 0, D_2 > 0, D_3 > 0, D_4 > 0, ...$$
$$\text{Local maximum: } D_1 < 0, D_2 > 0, D_3 < 0, D_4 < 0, ...$$

If neither of these requirements are met, the point is a saddle point. For the $(n+1) \times (n+1)$ Hessian matrix the two first corresponding determinants are

$$D_1 = 0$$
$$D_2 = -\Big(z_{n+1}\frac{\partial f_n^1}{\partial z_1 \partial z_2} + \frac{\partial f_n^2}{\partial z_1 \partial z_2}\Big)^2 < 0$$

D_2 is negative. Thus, all inner critical points are saddle points. The extrema must therefore be found in a lower dimension. In the n variable case, however, the minimum and maximum values are by assumption located in the corners.

It is true by induction that extrema are obtained in the corner points for all n in the generic situation.

With the extreme values identified, the constant of the production term for the $n \times n$ case can be determined. The process is analogous to the approach used in chapter 6. If m denotes the minimum value of f_n, the constant has to be grater than the absolute value of m in order to ensure a positive production term.

Chapter 10

Discussion

The aim of this thesis was to implement and analyse metamodels of typical gene regulatory networks.

A discrete approximation of the Hill function was established by principal component analysis. 5 principal components were needed to describe the function properly. Similar analysis has been performed previously in the PhD thesis of Julia Isaeva; the article "Nonlinear modelling of curvature by bi-linear metamodelling" suggests using 11 principal components when approximating the Hill function [4]. While the threshold value is considered constant in this thesis, the article also takes variable threshold value into account. In addition, the range of the steepness parameter is wider and the error is smaller. Thus, it is as expected that less components are needed in this metamodel - the result is consistent with previous observations.

When modelling circadian oscillations and the 3×3 system of genes, we observed deviations close to the limits of the model. The problem was solved by expanding the metamodel, included increasing the number of principal components. This is somewhat undesirable. For the efficiency, as few components as possible is advantageous.

On the other hand, two benchmark problems where implemented with acceptable results - indicating that the model is a good fit.

Chapter 11

Conclusion and Outlook

In this thesis metamodels of typical gene regulatory networks were implemented and analysed. Based on the study of the relative squared error, the approximation of the scalar case and the second order system was considered acceptable. For higher order systems it became necessary to expand the metamodel in order to model the systems properly.

The extreme values were investigated when determining the product terms of the genetic networks; in a generic situation it was proved that extrema are obtained in the corner points.

In both the PhD thesis of Thomas Mestl [12] and the article by Plathe and Kjøglum [16], stationary (steady) points and their stability are discussed. This could be a natural next step for further work.

Another possible development is to implement a general Matlab code for a $n \times n$ genetic network.

Bibliography

[1] *All Nobel Prizes in Chemistry.* URL: `https://www.nobelprize.org/nobel_prizes/chemistry/laureates/` (visited on 21/11/2016).

[2] Albert Goldbeter. "A Model for Circadian Oscillations in the Drosophila Period Protein (PER)". In: *Proceedings: Biological Sciences, Volume 261, Issue 1362.* (1995), pp. 319–324.

[3] Anthony Griffiths, Jeffrey Miller, and Davis Suzuki et al. *An Introduction To Genetic Analysis.* 6th edn. New York: W. H. Freeman and Company, 1996, pp. 383–412.

[4] Julia Isaeva. "Multivariate analysis as a tool for understanding and reducing complexity of mathematical models in systems biology". PhD thesis. Norwegian University of Life Sciences, 2011.

[5] Werner Kohler and Lee Johnson. *Elementary Differential Equations with Boundary Value Problems.* 2th edn. Virginia Tech: Pearson, 2006, pp. 89–98, 496–498.

[6] Erwin Kreyszig. *Advanced Engineering Mathematics.* 10th edn. Ohio State University: John Wiley & Sons, Inc, 2011, pp. 805–867.

[7] Alan Laub. *Matrix Analysis for Scientists & Engineers.* Arizona State University: Siam, 2005, pp. 139–142.

[8] David C. Lay. *Linear Algebra and Its Applications.* 4th edn. University of Maryland-College Park: Pearson, 2012, pp. 414–430.

[9] MathWorks. *interp1. 1-D data interpolation (table lookup).* URL: `http://se.mathworks.com/help/matlab/ref/interp1.html` (visited on 12/09/2016).

[10] MathWorks. *ode45. Solve nonstiff differential equations — medium order method.* URL: `http://se.mathworks.com/help/matlab/ref/ode45.html` (visited on 12/09/2016).

[11] Jagdish Mehra. *The Physicist's Conception of Nature.* Boston: D. Reidel Publishing Company, 1973, p. 618.

[12] Thomas Mestl. "Mathematical Approaches to Describe and Analyse the Dynamics of Gene Regulatory Systems". PhD thesis. Agricultural University of Norway, 1995.

[13] Thomas Mestl, Erik Plahte, and Stig Omholt. "A Mathematical Framework for Describing and Analysing Gene Regulatory Networks". In: *Journal of Theoretical Biology* (1995).

[14] Douglas C. Montgomery. *Design and Analysis of Experiments*. 8th edn. University of California: John Wiley & Sons, Inc, 2013, pp. 248–252.

[15] NDLA. *Proteinsyntesen.* URL: http://ndla.no/nb/node/47058?fag=7 (visited on 08/11/2016).

[16] Erik Plahte and Sissel Kjøglum. "Analysis and generic properties of gene regulatory networks with graded response functions". In: *Physica* (2004), pp. 150–176.

[17] Erik Plahte, Thomas Mestl, and Stig Omholt. "A methodological basis for description and analysis of systems with complex switch-like interactions". In: *Journal of Mathematical Biology* (1998), pp. 321–348.

[18] Arkadi Ponossov. *Lectures in Parameter Estimation Methods*. Norwegian University of Life Sciences, 2015.

[19] The Biology Project. *Rational Function Applications. Hill Equation.* URL: http://www.biology.arizona.edu/biomath/tutorials/rational/applications/hill.html (visited on 02/09/2016).

[20] Karl Erik Sandvold, Stein Øgrim, and Tone Bakken et al. *Gyldendals formelsamling i matematikk*. Gyldendal, 2008.

[21] Yuri Shtessel et al. *Introduction: Intuitive Theory of Sliding Mode Control.* URL: http://www.springer.com/cda/content/document/cda_downloaddocument/9780817648923-c1.pdf?SGWID=0-0-45-1391207-p175157910 (visited on 15/09/2016).

[22] Brian Sittinger. *The Second Derivative Test in n variables.* URL: http://faculty.csuci.edu/brian.sittinger/2nd_DerivTest.pdf (visited on 25/10/2016).

[23] George Thomas, Maurice Weir, and Joel Hass. *Thomas' Calculus*. 12th edn. Boston: Pearson, 2010, pp. 184–190, 802–808.

[24] William Trench. *Introduction to Real Analysis*. San Antonio: Pearson Education, 2003, pp. 11–12.

Appendix A

Matlab Codes

A selection of Matlab codes is presented in this appendix. Other codes can be obtained upon request.

A.1 Approximating the Hill Function

```
function[z1app, z2app] = approximateZ()
% Approximates the matrix H with 5 principal components
    for a fixed value of q
%   returns: the approximation of H for q=0.2 and q=0.4

    x0 = 0.01;
    xJ = 1.5;
    q0 = 0.1;
    qI = 0.5;
    I = 100;
    J = 150;

    hillMat = makeHillMatrix( x0, xJ, q0, qI, J, I);

    [U,S,V] = svd(hillMat);
    nComponents = 5;

    L = zeros(100,nComponents);

    for i = 1:nComponents
        L(:,i) = S(i,i)*U(:,i);
    end
```

```
    z1app = L(25,1)*V(:,1) + L(25,2)*V(:,2) + L(25,3)*V
        (:,3) + L(25,4)*V(:,4)+L(25,5)*V(:,5);
     % q = 0.2 is the 25'th value
    z2app= L(75,1)*V(:,1) + L(75,2)*V(:,2) + L(75,3)*V
        (:,3) + L(75,4)*V(:,4)+L(75,5)*V(:,5);
     % q = 0.4 is the 75'th value

end

function [ hillMatrix ] = makeHillMatrix( x0, xJ, q0,
    qI, J, I)
%Discretizing the Hill function
%   x0: lowest x value
%   xJ: highest x value
%   q0: lowest q value
%   qI: highest q value
%   J: number of x steps
%   I: number of q steps
%   returns: the matrix H containing the reduced Hill
    function values

    if x0 <= 0
        disp('x-value has to be positive')
        return
    end

    if q0 <= 0
        disp('q-value has to be positive')
        return
    end

     xValues = linspace(x0, xJ, J);
     qValues = linspace(q0, qI, I);

     hillMatrix = ones(I, J);

     for row = 1:I;
         for col = 1:J;
             hillMatrix(row, col) = hillReduced(xValues
                 (col), qValues(row));
```

```
            end
        end

end


function [ value ] = hillReduced( x,q )
%Hill function with treshold value equal to 1
%   x: gene consentration
%   q: steepness parameter
%   returns: Hill function value

    value = hill(x,q,1);

end


function [ value ] = hill( x,q,theta )
%Hill function
%   x: gene consentration
%   q: steepness parameter
%   theta: treshold value
%   returns: Hill function value

    value = (x.^(1/q))/(x.^(1/q) + theta.^(1/q));

end
```

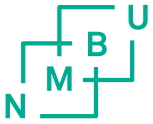
A.2 Solving Systems of Differential Equations

Of practical considerations, only the code solving the 2x2 system of genes (chapter 6) is included. The method is equivalent for systems of higher and lower order.

```
function [t,y] = odeSystemMetamodel(a1,a2,b1,b2,c1,c2,
    gamma1,gamma2,x01,x02)
%Solves system of 2 ODEs with interpolation of the Hill
    function
    %a1,a2,b1,b2,c1,c2,gamma1,gamma2,x01,x02:
        coefficients
```

```
    %returns: time interval and corresponding solution
        of x

    [z1app, z2app] = approximateZ();

    xj = linspace(0.01,1.5,150);
    F1 = @(xq)interp1(xj,z1app,xq, 'linear', 'extrap');
    F2 = @(xq)interp1(xj,z2app,xq, 'linear', 'extrap');

    tspan = [0 1.5];
    initialValues = [x01 x02];
    f = @(t,x) [a1*F1(x(1)).*F2(x(2))+b1*F1(x(1))+c1*x
        (2)+d1-gamma1*x(1); a2*F1(x(1))*F2(x(2))+b2*F1(x
        (1))+c2*x(2)+d2-gamma2*x(2)];

    [t,y] = ode45(f, tspan, initialValues);

end

function [ t,y ] = odeSystemOde45(a1,a2,b1,b2,c1,c2,
    gamma1,gamma2,x01,x02)
%Solves system of 2 ODEs
    %a1,a2,b1,b2,c1,c2,gamma1,gamma2,x01,x02:
        coefficients
    %returns: time interval and corresponding solution
        of x

    if (d1< analyticProductionTest(a1,b1,c1) | d2<
        analyticProductionTest(a2,b2,c2))
         'productioin term is negative, choose new
            values for d'
         return
    end

    tspan = [0 1.5];
    initialValues = [x01 x02];
    f = @(t,x) [a1*z1(x(1))*z2(x(2))+b1*z1(x(1))+c1*x
        (2)+d1-gamma1*x(1); a2*z1(x(1))*z2(x(2))+b2*z1(x
        (1))+c2*x(2)+d2-gamma2*x(2)];

    [t,y] = ode45(f, tspan, initialValues);
```

```
end

function [ z1value ] = z1(x)
%Hill function with q=0.2
%    returns: hill function value

    z1value = power(x,(1/0.2))/(power(x,(1/0.2))+1);

end

function [ z2value ] = z2(x)
%Hill function with q=0.4
%    returns: hill function value

    z2value = power(x,(1/0.4))/(power(x,(1/0.4))+1);

end

function [setD] = analyticProductionTest(a,b,c)
%finds the minimum value of the coefficient d
%to ensure a positive production term
    %a: coefficient
    %b: coefficient
    %c: coefficient
    %returns: the mimimum value of the coefficent d

    boundaryValues = [c b 0 a+b+c];
    setD = abs(min(boundaryValues));

end
```



Norges miljø- og biovitenskapelig universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway