UPTEC X 03 031
NOV 2003
ISSN 1401-2138

JONAS SÖDERBERG

Global regulation of bacterial growth

Master's degree project

UPPSALA UNIVERSITET

Molecular Biotechnology Programme

Uppsala University School of Engineering

UPTEC X 03 031	**Date of issue 2003-12**

Author

Jonas Söderberg

Title (English)

Global regulation of bacterial growth

Title (Swedish)

Abstract

Bacteria growing in different media adjust to these in order to grow as fast as possible. This is regulated by a complex network of control systems. I have modelled the cell's response to amino acid deficiency in the medium. The response to amino acid starvation is called "the stringent response" and causes amino acid production to increase in the cell in order to cover for the starvation. The model started out from the fact that the level of ppGpp (guanosine tetraphosphate) in the cell increases drastically during the stringent response. A mathematical model for the role of ppGpp in the response was created. The results showed that the concentration of ppGpp indeed is a possible control mechanism for the stringent response to nutritional stress.

Keywords

ppGpp, control theory, stringent response

Supervisors

Måns Ehrenberg
Uppsala University

Scientific reviewer

Kurt Nordström
Uppsala University

Project name	Sponsors
Language **English**	Security
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **50**

Biology Education Centre Biomedical Center Husargatan 3 Uppsala
Box 592 S-75124 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 555217

Global regulation of bacterial growth

Jonas Söderberg

Sammanfattning

Bakterieceller som växer i olika medier anpassar sig till dessa så att de kan växa så snabbt som möjligt under olika förhållanden. En bakterie som växer i ett fattigt medium, utan tillgång till vare sig aminosyror eller en fördelaktig kolkälla som t ex glukos, måste tillverka fler typer av enzymer än en bakterie som växer i ett rikt medium. Detta gör att maximal tillväxthastighet i ett fattigt medium är lägre än maximal tillväxthastighet i ett rikt medium, och följaktligen kräver färre ribosomer per cellmassa. Eftersom tillgången på aminosyror och kolkällor i omgivningen snabbt kan förändras, behöver bakterien snabbt kunna reglera sin ribosommängd, vilket görs genom en kombination av olika reglersystem. Jag har konstruerat en förenklad matematisk modell av hur denna reglering kan gå till.

Modellen är uppbyggd kring molekylen guanosintetrafosfat (ppGpp), vars koncentration korrelerar negativt med aminosyratillgången i cellen i relation till ribosomernas förmåga att konsumera dem i proteinsyntesen. Koncentrationen av ppGpp styr syntesen av nya ribosomer genom att påverka ett antal reglermekanismer. För att kunna testa modeller för dessa reglersystem, programmerade jag systemet i MATLAB och gjorde en simulering av hur koncentrationerna i cellen förändras med tiden då aminosyratillgången i mediet plötsligt ändras.

Examensarbete 20 p i Molekylar bioteknikprogrammet

Uppsala universitet December 2003

Contents

1 Introduction

Bacteria use various control mechanisms in order to sustain maximal growth rate in a varying environment. If, for instance, lack of one amino acid in the medium would occur, the bacterium uses intrinsic systems of control to make sure that the amino acid will be produced. How these systems are connected is poorly understood and very few models successfully predict the behaviour of the integrated physiology of a whole cell. The only knowledge lies within the local control mechanisms and not in global regulation. The major quest that lies ahead is to combine existing models and design new ones with the purpose to gain understanding of how the intricate global control system that nature created works.

1.1 Overview

This report starts by introducing the background to the system which I have simulated. This is done by showing the intricate biology that lies behind the model. Furthermore, I will tell you about the model itself, the mathematics involved and which approximations I had to make in order to get everything to work. After that I will say something about the implementation of the model using MATLAB and Maple and show some results and explain what is implied by those. Finally I will discuss the results and how they can be interpreted and improved in the future.

1.2 Background

Bacteria that grow in different media have different needs for amino acid producing enzyme systems [7]. In a rich medium, the level of amino acid production does not need to be great, but during amino acid starvation the missing amino acid must be produced in order for the cell to continue living. In wild type *E. coli* "The stringent response" is the response to stress caused by drops in the amino acid levels [3]. The stringent response is triggered by the molecule *guanosine tetraphosphate (ppGpp)*, and the way by which this process is regulated is dependent on the concentrations of many different nutritients and enzymes. Microbiologists have for a very long time been interested in deciphering the logic of the network of the control mechanisms involved in cellular growth. The way to see this is to break down the system into smaller bits and see how all of these work in isolation. However, in nonlinear systems the sum of the parts does not describe the behaviour of the whole and it is therefore important to describe as large a system as possible. What I have done is to introduce some new elements in the modelling of the stringent response in order to increase the relevance of the model of the cell.

1.3 Amino acid metabolism and growth control in *E. coli*

Among the various control systems present in the bacterium I have chosen to study the perhaps most important parameter to the cell; the growth rate. In

order to find out how the cell maximizes its growth I had to model the various subsystems involved in cellular growth and connect them. Since this is beyond the scope of this report I chose to study one selected property of cellular growth, namely its relation to the stringent response.

1.3.1 The stringent response

The stringent response is the physiological response induced by the failure of tRNA aminoacylation to keep up with the demand for aminoacyl-tRNA from protein synthesis. This condition may be implemented by either limiting the supply of amino acids, or by reducing the ability to aminoacylate tRNA. Many of the processes initiated during the stringent response behave as if being mediated by accumulation of ppGpp.

The first effect when the stringent response is induced [3] is a rapid inhibition of the accumulation of stable RNA (*rRNA* and *tRNA*). In mutant cells lacking the ability to accumulate ppGpp due to a mutation in the *relA* gene, the stringent response to amino acid starvation is absent. Since no particular measures are taken to counter the amino acid starvation, this mutational response is designated *the relaxed response*.

1.3.2 Guanosine tetra phosphate (ppGpp)

Guanosine Tetra Phosphate is a derivative of the molecule GDP and the main effector molecule of the stringent response[8]. It is mainly produced in two ways in the cell [3]; by the two enzymes ppGpp synthetase one (PSI) and ppGpp synthetase two (PSII). These two enzymes are in turn expressed from the genes *relA* and *spoT*. The ppGpp synthetases produced from the genes *relA* and *spoT* are also often called RelA and SpoT as well as PSI and PSII. There is also the molecule *guanosine penta phosphate* (pppGpp). It is produced in the same way as ppGpp and has the same role in the stringent response. Thus, I treat them as being the same molecule for modelling purposes. The pathways of producing and converting ppGpp can be viewed in Figure 1. The activity of PSI is regulated by the numbers of deacylated cognate tRNAs located at a ribosome's A site. The gene *relA* is constitutively expressed, and thus the control of the level of ppGpp lies in the activity of PSI. The gene *spoT* which codes for PSII also codes for an enzyme responsible for breaking down ppGpp. Which gene product is produced may depend on the velocity by which the ribosome incorporates amino acids and thus on the level of charged tRNA [6].

1.3.3 tRNA

The transfer RNA consists of stable RNA folded into a "clover leaf"-like structure. One side of the tRNA structure consists of three nitrogen bases coding for the corresponding specific amino acid and the other is the docking site for the amino acid in question as can be seen in Figure 2. The ribosome incorporates amino acids into a nascent protein by first binding a cognate (the tRNA is successfully paired with the codon on the mRNA where the ribosome is at

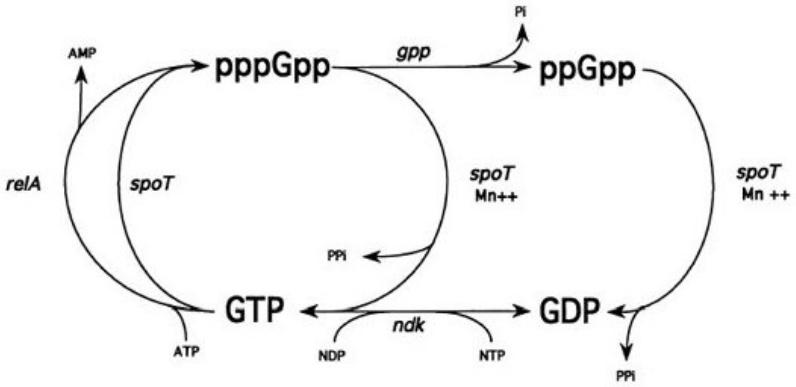


Figure 1: Cellular routes of (p)ppGpp metabolism. The enzymes involved in (p)ppGpp metabolism are represented by their respective structural genes. The interconversions shown are thought to occur under physiological conditions, although there are additional activities not shown and not all shown are used. For example, the RelA enzyme can convert GDP to ppGpp as easily as GTP to pppGpp. However, the physiological levels of GDP are small relative to the levels of GTP and therefore this route of ppGpp synthesis is not shown. The enzymes for ppGpp metabolism that are now well defined are PSI (*relA*), (p)ppGpp 5'-phosphohydrolase (*gpp*), (p)ppGpp 3'-pyrophosphohydrolase (*spoT*), and nucleoside 5'-diphosphate kinase (*ndk*). *Illustration used with permission from Michael Cashel*

the moment) tRNA. The ribosome then adds the amino acid brought by the tRNA to the elongating protein. The tRNA only carries one amino acid and needs to be recharged in order for the ribosome to use it again. This is done by synthetases that take free amino acids and attach them covalently to the correct tRNA [5].

1.3.4 The ribosome

The ribosome is an enzymatic system consisting of three stable RNA (rRNA) chains and about 50 proteins. These form two subunits, the large (50S) and the small (30S) subunit. The ribosome is located in the cytosol and its main purpose is to synthesise proteins by catalysing formation of peptide bonds between amino acids in sequences determined by the reading of the mRNA code by tRNA molecules. The ribosome moves across the mRNA, and thereby making different triplets of bases on the mRNA available for reading by tRNA. Each triplet codes for one amino acid and sets up a specific binding site for the cognate tRNA. In order to be able to bind to mRNA the ribosome also includes RNA of its own, the so called rRNA. Since the velocity of protein synthesis depends on the availability of the correct amino acids, the ribosome produces proteins at a rate that is limited by the concentrations of cognate aminoacylated tRNAs.

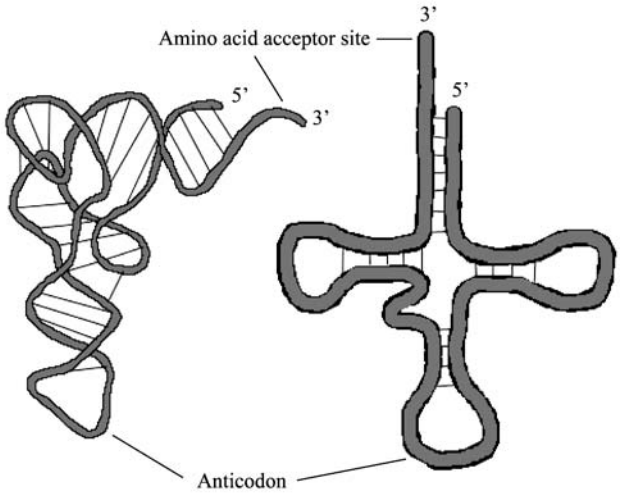


Figure 2: All tRNA are single-stranded molecules that fold in three loops joined by short stems consisting of the paired nitrogenous bases. The "clover leaf" structure of tRNA thus formed is seen to the right. In three dimensions, tRNA is twisted to make an "L" shape; the middle loop forms one leg of the "L" and the 3' and 5' ends make the other leg, as seen to the left. *This figure is modified from Figure 16.10 in Biology 4th edition [2]*

1.3.5 Enzymes

The enzymes in the cell are normally proteins and hence produced by the ribosome. The enzymes catalyse formation of e. g. amino acids and effector molecules. They are often degenerated in the cell by the work of other enzymes and the amino acids they consist of are thereby returned to the cytosol. When enzymes are produced in *E. coli*, several ribosomes use the same molecule of mRNA and the translation begins as soon as the promoter on the mRNA is transcibed. Enzymes produce amino acids by synthesis of new ones and decay of old proteins. The ribosome assembles the amino acids into proteins using tRNA, including new ribosomal proteins and enzymes. This implies that the concentrations of amino acids, enzymes and ribosomes are depending on each other. I. e. the regulation of their concentrations needs to be very precise to keep the concentrations in the cell at a physiological level.

1.3.6 Amino acids

Amino acids are molecules consisting of an amino group, a carboxyl group and a side chain. The side chains determine the identities of the different amino acids.

1.3.7 RNA polymerase

For the cell to be able to transcribe DNA, translate mRNA, bind to nucleic acids complementarily and many other tasks involving nucleic acids, the cell needs different kinds of RNA. The process of transcribing RNA from DNA is performed by the RNA polymerase (Abbreviated *RNAP*). The RNAP binds to a promoter on the DNA and from there it starts to elongate the chain of bases, in the order given by the complementary DNA. The transcribed RNA is then processed by other enzymes if needed before usage.

1.4 Intentions

My intention with this project was to simulate the behaviour of a cell during nutritional stress induced by an amino acid deficiency in the surrounding medium. In the analysis I included the time delays between initiation of transcription of a gene and the emergence of an active protein. The models analysed in this work include features not previously treated in the literature.

2 The Model

When modelling entire cells, a natural starting point is the exponential growth rate μ:

$$\mu = \frac{r \cdot v}{\rho_0} \tag{2-1}$$

Here r is the concentration of elongating ribosomes, v is the average protein elongation rate per ribosome and ρ_0 is the total concentration of amino acids contained in proteins in the cell. $r \cdot v$ is therefore the total rate of synthesis of peptide bonds per volume, and when $r \cdot v$ is normalised to ρ_0 one obtains the relative mass increase per time unit, synonymous to μ.

2.1 Setting up the system

Expanding the model from 2-1 I introduced the following expression for ρ_0:

$$\rho_0 = \Omega + 1000 \cdot e + 10000 \cdot r + 1000 \cdot RNAP_{tot} \tag{2-2}$$

In this expression Ω is the concentration of amino acids contained in housekeeping enzymes. I. e the proteins that are not explicitly present in the model but still contain amino acids. e is the total concentration of enzymes that *are* explicitly modelled, r is the concentration of ribosomes in the cell and $RNAP_{tot}$ is the total concentration of RNA polymerase in the cell.

One assumption is that degradation of proteins and decay of mRNA occur with rates d and λ, respectively, and that the degradation of proteins releases free amino acids into the cell. Further, I assumed that there are 20 amino acids produced by 20 enzymatic systems. These were in turn produced from amino acids by the ribosome according to the mRNA for the enzyme system produced by RNA polymerase.

2.2 The ribosome

The production of ribosomes is limited by the concentration of either rRNA or ribosomal proteins in the cell. By assuming that the assembly into a ribosome of both proteins and rRNA is instantaneous and that the regulation of the production of rRNA does not depend on the concentration of ribosomal proteins, the production of ribosomal proteins (R below) can be treated as being regulated through a binary switch. If rRNA is limiting, no ribosomal proteins are produced. If the ribosomal proteins are limiting, their production becomes maximal instantaneously. The concentration of ribosomes is thus determined by $r = \min(R, rRNA)$. The changes in concentrations for rRNA and ribosomal proteins can be expressed as:

$$\frac{d\text{rRNA}}{dt} = P_{1prod} + P_{2prod} - \mu \cdot rRNA \tag{2-3}$$

$$\frac{dR}{dt} = \frac{mRNA_R \cdot r}{K_r + r} \tag{2-4}$$

In 2-3, P_{1prod} and P_{2prod} are the rates of production of rRNA made by the RNAP from the promoters P1 and P2, respectively [1]. P1 and P2 are the two kinds of promoters that codes for stable RNA [1], as explaineded in section 2.3. In 2-4, $mRNA_R$ is the concentration of mRNA coding for ribosomal protein as calculated in 2-10. K_r is the K_m-value (Michaelis-Menten) for the binding of a ribosome to mRNA. Since the α in the expression for $mRNA_R$ in 2-4 is either one or zero and synthesis of ribosomal proteins is down-regulated when these are in excess over rRNA, it is on these promoters P1 and P2 that the active regulation of the concentration of ribosomes lies.

The regulation of the concentration of ribosomes is the key to preventing excessive use of amino acids during starvation and also to assure that a sufficient amount of enzymes is produced. In my model, the regulation of the ribosome is mainly due to changes in the concentration of ppGpp, which affects the rate of ribosome production by forming a complex with RNAP and thus inhibiting its binding to the promoters for stable RNA (P1 and P2). By assuming that the equilibrium between ppGpp and RNAP is fast, one gets the following expression for the concentration of complex:

$$ppGpp + RNAP_f \underset{K_{rp}}{\rightleftharpoons} RNAP_{ppGpp} \tag{2-5}$$

In this scheme $ppGpp$ is the concentration of ppGpp in the cell, $RNAP_f$ is the concentration of all free functional RNA polymerase and $RNAP_{ppGpp}$ is the concentration of the fraction of free functional RNA polymerase found in a complex with ppGpp.

$$\begin{aligned} RNAP_{ppGpp} = \frac{1}{2} \cdot (ppGpp + RNAP_f + K_{rp} - \\ - \sqrt{(ppGpp + RNAP_f + K_{rp})^2 - 4 \cdot ppGpp \cdot RNAP_f}\Big) \end{aligned} \tag{2-6}$$

In this equation, K_{rp} is the dissociation constant for the complex between RNAP and ppGpp. Furthermore, the two promoter types P1 and P2 produce stable RNA according to

$$\begin{aligned} P_{1prod} &= P_1 \cdot V_{1max} \cdot \left(\frac{1}{1 + \frac{K_{m1}}{RNAP_f(1-q)}} + \frac{1}{1 + \frac{1000K_{m1}}{RNAP_f \cdot q}} \right); \qquad q = \frac{RNAP_{ppGpp}}{RNAP_f} \\ P_{2prod} &= P_2 \cdot V_{2max} \cdot \left(\frac{1}{1 + \frac{K_{m2}}{RNAP_f(1-q)}} + \frac{1}{1 + \frac{1000K_{m2}}{RNAP_f \cdot q}} \right) \end{aligned} \tag{2-7}$$

Since the affinity K_m for the promoters is changed, the parantheses include two separate terms. The first one using the pure RNAP and the socond one using the ppGpp bound RNAP. The production of mRNA for constitutively expressed proteins follows the expression

$$P_{3prod} = P_3 \cdot V_{3max} \cdot \left(\frac{1}{1 + \frac{K_{m3}}{RNAP_f}} \right) \tag{2-8}$$

In these equations P_i is the concentration of the promoter of type i, K_{mi} is the halfmax concentration of RNAP for promoter i (Michaelis-Menten) and V_{imax} is the maximal activity of the RNAP in initiations per promoter i and time unit.

2.3 RNA polymerase

The concentrations of the RNAs can be viewed as a flow originating from the RNA polymerase (RNAP), but we still would have to determine that flow. One can divide RNA promoters into five groups [1] with different regulatory systems and different tasks in the cell. There are P1 and P2 which promote the production of stable RNA with variable and constant V_{max} and K_m respectively, where V_{max} is the maximal RNA chain elongation velocity in bases per second for RNAP, and K_m is the Michaelis-Menten parameter for the half-max concentration of RNAP [1]. There is also P3 which is a strong constitutive mRNA promoter and P4 which is a weak repressible mRNA promoter [1]. Last, there is P5 which denotes the pause sites of the RNAP. This P5 can be seen as a ppGpp-dependent constitutive promoter that gives rise to time delays in the production of mRNA [1].

In order to generalise this expression, I took the promoter concentrations (copy number per volume) for the five different types taken at two different growth rates and, assuming that the promoter concentration P_i was only dependent on the growth rate and could be written on exponential form, I expressed the concentrations of the promoters as:

$$P_i = A_i^{\frac{\mu}{B_i}} \qquad i = 1..5 \tag{2-9}$$

The above expression was deduced with different constants A and B for the five types which leads to the forming of the following expression for RNA:

$$RNA_i = \frac{\frac{\alpha_i P_i V_{imax}}{1+\frac{K_{mi}}{RNAP_f}}}{\lambda + \mu} \qquad i = 1..5 \tag{2-10}$$

In this expression, RNA_i is the concentration of RNA from promoter type i, α_i is a regulation factor which can be a constant or be regulated depending on the promoter class, P_i is the promoter concentration as calculated in 2-9, V_{imax} and K_{mi} are the speed and the half-max concentration (Michaelis-Menten) of RNAP respectively, $RNAP_f$ is the concentration of free functional RNAP, and λ and μ are the decay and dilution of RNA, respectively.

2.4 Guanosine tetra phosphate (ppGpp)

The core of my model is the effector molecule ppGpp. As seen in section 2.2 it is regulating the concentration of ribosomes and thus the entire growth of the cell as defined in section 2.1. ppGpp is synthesised in two ways; either by PSI (RelA) from the *relA*-pathway, or by PSII (spoT) from the *spoT*-pathway. In both ways ppGpp is produced from GTP. Further, the concentration is decreased by the enzymatic degradation caused by another product of the *spoT*-pathway and by dilution. This led to the following schematics:

$$GTP \underset{\text{hydrolases and dilution}}{\overset{\text{synthetases}}{\rightleftharpoons}} ppGpp \tag{2-11}$$

Since the ppGpp synthetases are working according to a zero order reaction, assuming a fixed concentration of GTP, the concentration of ppGpp only depends

on μ and the concentrations of PSI, PSII and ppGpp hydrolase. This ppGpp hydrolase is also a product of the *spoT*-pathway, but unlike PSII it breaks down ppGpp (See section 2.4.2 below). This may seem easy, but all of these depend on many other systems. Nevertheless, the differential equation for producing ppGpp is

$$\frac{dppGpp}{dt} = \frac{PSII \cdot k_{PSII}}{1 + \frac{ppGpp}{K_m^{PSII}}} + \frac{PSI \cdot k_{PSI}}{1 + \frac{ppGpp}{K_m^{PSI}}} - \frac{hydrolase \cdot k_{hydrolase}}{1 + \frac{K_m^{hydrolase}}{ppGpp}} - \mu \cdot ppGpp \tag{2-12}$$

In this equation k_{PSI} and k_{PSII} are the constant velocities by which PSI and PSII produce ppGpp respectively, PSI and $PSII$ are the concentrations of *active* ppGpp-synthetases, K_m^i is the K_m-value (Michaelis-Menten) for the enzyme i and *hydrolase* is the concentration of ppGpp hydrolase. This deceivingly simple equation will now be expanded as we include the expressions for PSI and PSII and their respective control systems.

2.4.1 RelA

PSI is active only when bound to the ribosomal A site together with deacylated tRNA cognate for the A site codon [10]. Therefore, the production of ppGpp depends on the level of cognate and deacylated tRNA located at a *translating* ribosome's A site. That means that if there is a shortage of one amino acid, there will be a high concentration of its deacylated tRNA. This, in turn, means that there is a greater probability that the A site will be occupied by a deacylated tRNA, which implies that the production of ppGpp will be higher. In order to calculate this production I assumed that the level of total PSI is constant, and that the productive PSI-tRNA^c-A site-complex (tRNA^c denotes a *cognate* tRNA) is in equilibrium according to the following scheme:

$$\begin{aligned} A + R + T &\underset{k_{-a}}{\overset{k_a}{\rightleftharpoons}} AR + T \underset{k_{-t}}{\overset{k_t}{\rightleftharpoons}} ART \\ A + R + T &\underset{k_{-t}}{\overset{k_t}{\rightleftharpoons}} AT + R \underset{k_{-b}}{\overset{k_b}{\rightleftharpoons}} ART \end{aligned} \tag{2-13}$$

In this, A is the concentration of free unoccupied A sites (E. g. mRNA bound ribosomes without RelA (PSI) or tRNA), R is the concentration of free RelA (PSI), and T is the concentration of free deacylated tRNA^c. Assuming that the system equilibrates fast with given constants, the following expression can be

solved for the active complex ART:

$$\begin{cases} ART = \dfrac{k_t \cdot AR \cdot T}{k_{-b} + k_t} + \dfrac{k_b \cdot AT \cdot R}{k_{-b} + k_t} \\ AR = \dfrac{k_a \cdot A \cdot R}{k_{-a} + k_t \cdot T} + \dfrac{k_{-t} \cdot ART}{k_{-a} + k_t \cdot T} \\ AT = \dfrac{k_t \cdot A \cdot T}{k_{-t} + k_b \cdot R} + \dfrac{k_{-b} \cdot ART}{k_{-t} + k_b \cdot R} \\ A = \dfrac{k_{-a} \cdot AR}{k_a \cdot R + k_t \cdot T} + \dfrac{k_{-t} \cdot AT}{k_a \cdot R + k_t \cdot T} \\ K_t = \dfrac{k_{-t}}{k_t} \\ K_a = \dfrac{k_{-a}}{k_a} \\ K_b = \dfrac{k_{-b}}{k_b} \\ A_0 = ART + AR + AT + A \\ R_0 = ART + AR + R \\ T_0 = ART + AT + T \end{cases} \tag{2-14}$$

In this formula A_0 is the *total* concentration of A sites located on mRNA bound ribosomes (E. g. all mRNA bound ribosomes without acylated tRNA), T_0 is the *total* concentration of deacylated tRNA^c and R_0 is the *total* concentration of RelA in the cell. K_t, K_a and K_b are the dissociation constants for the binding of tRNA^c to the ribosome, PSI to an A site and PSI to a free A site with a bound tRNA^c, respectively.

The concentration of free A sites (A_0) of translating ribosomes is dependent on the concentration of ribosomes that are bound to mRNA (Those who are translating) and the level of charged tRNA^c according to the formula

$$A_0 = B_R - \left(\frac{1}{2} \left(B_R + \sum_{n=1}^{20} \left(f_n \cdot T_n^3 \right) + 10^{-7} - \right. \right. \\ \left. \left. - \sqrt{\left(B_R + \sum_{n=1}^{20} \left(f_n \cdot T_n^3 \right) + 10^{-7} \right)^2 - 4 \cdot B_R \sum_{n=1}^{20} \left(f_n \cdot T_n^3 \right)} \right) \right), \tag{2-15}$$

where B_R is the concentration of all ribosomes bound to mRNA, f_n is the fraction, normalised to one, by which the n:th amino acid is used in protein synthesis and T_n^3 is the concentration of charged tRNA (I. e. ternary complex, a tRNA^c with an amino acid, elongation factor EF-Tu and GTP bound to it.).

Since f_n is the only known constant, we want to get an expression for B_R and T_n^3. The number of bound ribosomes can be obtained from the total ongoing translation of all enzymes and proteins. The expression obtained will be

$$B_R = \frac{C + 1000 \sum_{n=1}^{20} (e_n) + 10000 \cdot r}{1000 \sum_{n=1}^{20} (e_n) + 10000 \cdot r} \left(\sum_{prod} \left(\sum_{n=1}^{20} \left(e_n + 50 \cdot r \right) \right) \right) \tag{2-16}$$

This is true since the pool of housekeeping proteins (C) is near constant and known, and since the lengths of the transcripts is known. In 2-16, e_n is the total concentration of the n:th enzyme system, r is the total concentration of ribosomes in the cell and the sum over *prod* means the sum of all proteins currently being translated by the ribosome. Also, all enzymes are assumed to be 1000 amino acids long and the ribosome is assumed to consist of 50 proteins, each with a length of 200 amino acids.

As the total concentration of tRNAs is fairly constant, the concentrations for the ternary complexes (T_n^3) can be assumed to be equal to that total concentration for all tRNAs except for the one carrying the limitting amino acid, i. The concentration of ternary complex for the limitting amino acid can instead be obtained from the time it takes τ_i for the ribosome to incorporate amino acid i into a nascent protein. By using the expression for τ_i from 2-23, one get the following expression for the concentration of T_n^3:

$$T_n^3 = \frac{K_m}{\tau_i \cdot k_{ri}} \tag{2-17}$$

In this expression, K_m is the K_m-value (Michaelis-Menten) for the ribosomal usage of amino acids and k_{ri} is the maximal velocity of the ribosome in reactions per second.

Since PSI(RelA) only is active when bound to a ribosomal A site with a cognate tRNA, the active complex ART from 2-14 is used as PSI in 2-12. We still need to calculate PSII, though.

2.4.2 SpoT

The product of *spoT* has different activity if translated during stringent response or during abundance in the cell. Both PSII and the ppGpp hydrolase are presumably expressed from *spoT* and the model has to account for when one or the other is produced. According to our model, the only difference between the two proteins is the folding, and the different foldings are due to the concentration of amino acids in the cell. If there is an abundance of amino acids in the cell, the ribosome will work at maximal speed. The product of the gene *spoT* would in that case be ppGpp hydrolase, which degrades ppGpp. On the other hand, if there are few free amino acids in the cell, the ribosome will elongate the proteins more slowly, and thus the protein in production will have time to fold in an unfinished state. The prematurely folded protein will not dissemble ppGpp as expected, but instead *produce* it. In our model, this enzyme is called PSII or spoT, and the protein finished during amino acid abundance is called ppGpp hydrolase. The way to separate these two is to introduce a probability P for either one being produced depending on the velocity v of the ribosome. This leads us to the following expression for the probability to produce either ppGpp hydrolase or PSII:

$$\begin{aligned} P(\text{ppGpp hydrolase produced}) &= \frac{v^k}{C + v^k} \\ P(\text{PSII produced}) &= \frac{C}{C + v^k} \end{aligned} \tag{2-18}$$

In these probabilities, the numbers C and k are arbitrary constants chosen to make the model realistic and the decision surface fairly distinct. The decision of whether to produce the synthetase or the hydrolase is assumed to be done at the instant at which the ribosome finishes translating the mRNA from *spoT* and the product emerges in the cell. In every other way PSII and the hydrolase are both treated as other enzymes and thus being produced in the same way as shown in section 2.5. This will in this case be

$$\left[\frac{d\text{spoT}}{dt}\right]_{\text{init at time } t} = \frac{mRNA_{spoT} \cdot r}{K_r + r} - \lambda_{spoT} \cdot spoT \tag{2-19}$$

where the left side is the rate of initiation of translation of mRNA_{spoT} at time t, $mRNA_{spoT}$ is the concentration of the mRNA_{spoT}, K_r is a constant measuring the affinity the ribosome has for the mRNA, r is the concentration of ribosomes and λ_{spoT} is the rate of decay of spoT. The rate of finished protein at time t+T is then

$$\left.\begin{aligned}\left[\frac{d\text{PSII}}{dt}\right]_{t+T} &= P(\text{PSII produced})\left[\frac{d\text{spoT}}{dt}\right]_t \\ \left[\frac{d\text{hydrolase}}{dt}\right]_{t+T} &= P(\text{ppGpp hydrolase produced})\left[\frac{d\text{spoT}}{dt}\right]_t\end{aligned}\right\}, \tag{2-20}$$

where the left side is the rate of production of PSII (spoT) and ppGpp hydrolase at time t+T, respectively. The right side in 2-20 is the probability times the rate of production initiated at time t. The time T is the time it takes to finish translating a protein.

Since these equations are *not* equilibrating fast they have to be solved in every time step, which is done numerically. The answers to the equations are then used in 2-12.

2.4.3 Putting them all together

Since the concentration of ppGpp adjusts very fast to the level of nutritional stress compared to the proteins in my model (separation of time scales), one gets the condition that $dppGpp/dt = 0$. Insertion of the expressions for PSI and PSII in 2-12 then makes it possible to solve the equation. By solving equation 2-12, it was possible to get an explicit expression for the concentration of ppGpp as shown in Appendix C.

2.5 Biosynthetic enzymes

As mentioned before, the rate of initiation of translation of the mRNAs for the different enzymes changes over time according to the differential equation

$$\frac{de_i}{dt} = \frac{mRNA_i r_f}{K_r + r_f} - \lambda_i e_i, \tag{2-21}$$

where e_i is the concentration of N-terminal enzyme i, $mRNA_i$ is the concentration of the corresponding mRNA, K_r is the K_m-value (Michaelis-Menten) for

the binding of the ribosome to the mRNA, r_f is the concentration of free ribosomes and λ_i is the rate of decay of the enzyme i. For the *spoT*-enzymes (PSII and ppGpp hydrolase) and the ribosomal proteins the α in 2-10 is a constant value. For the amino acid producing enzymatic systems to be regulated properly, the α needs to be an expression depending on the translational velocity v of the ribosome. Assuming that the translation for all enzymes is regulated by ribosome mediated transcriptional attenuation or similar devices, one can calculate v using the following expression:

$$\begin{aligned} v &= \min_{i=1..20}\left(\frac{(k_i \cdot e_i + f_i \cdot \nu + \lambda \cdot f_i \cdot \rho)}{r \cdot f_i}, k_{ri}\right) \\ \text{where} \rho &= 1000 \cdot \sum_{j=1}^{20} e_j + 10000 \cdot r + 1000 \cdot RNAP_{tot} \end{aligned} \tag{2-22}$$

In this equation k_i is the rate by which enzyme i synthesises amino acid i, e_i is the concentration of enzyme i, f_i is the frequency of amino acid i in the proteins of the cell, ν is the inflow of amino acids from the medium, λ is the rate of decay of proteins, r is the concentration of ribosomes, $RNAP_{tot}$ is the total concentration of RNA polymerase and k_{ri} is the maximal rate of translation of the codon for amino acid i.

Equation 2-22 also assumes that it is only one amino acid that is limiting the whole protein synthesis. Further, 2-22 leads to the following expression for the time τ_i in seconds it takes to incorporate amino acid i into a polypeptide:

$$\begin{aligned} \tau_i &= \frac{1}{k_{ri}} && i = 1..20 \quad i \neq j \\ \tau_j &= \frac{1}{v \cdot f_j} - \frac{1 - f_j}{f_j \cdot k_{rj}} && j \text{ is the index of the limiting amino acid.} \end{aligned} \tag{2-23}$$

Now we are finally ready to calculate the regulating α:s used in 2-10. The resulting formula will then be:

$$\alpha_i = 1 - e^{-\left(\tau_i - \frac{1}{k_{ri}}\right)} \qquad i = 1..20 \tag{2-24}$$

This α is the factor which regulates the level of an amino acid through the production of the enzyme that produces it.

2.6 Time delays

During amino acid starvation, the ribosome will work slowly. However, transcription will be very fast compared to translation. This means that the time from initiation of transcription to the finished protein will be approximately as long as the translation time. To deal with this, time delays are introduced. A protein with length X will take X/v seconds to produce. During this time the nascent protein is inactive. This leads to a system of equations that is very stiff (sensitive to changes in the variables) and since v is varying, the time delays will change. The time it actually takes for RNAP to complete the promoter on the mRNA is considered to be a small constant factor not affecting the time

delays. This leads to the conclusion that only the delay from the ribosome needs to be modelled in order to get accurate results. This assumption also leads to the exclusion of explicit RNAP from the model since it doesn't do anything significant anymore. All the flows of mRNA are assumed to emerge from an unregulated constant pool of free RNAP.

2.7 Amino acids

The free amino acid concentrations are reduced by dilution through cellular growth or by protein synthesis. In protein synthesis, amino acids are first linked to their cognate tRNAs, forming ternary complexes with elongation factor EF-Tu and GTP. These ternary complexes are then bound to ribosomes programmed with codons matching the anticodons of the tRNAs. The process by which amino acids enter the pool is through production from the enzymatic systems, or from decay of existing proteins. The process can be simulated by the equation

$$\frac{dX_i}{dt} = \frac{k_i \cdot e_i}{1 + \frac{X_i}{K_p}} - v \cdot r \cdot f_i - \mu \cdot X_i + \frac{f_i \cdot \Phi_i}{1 + \frac{X_i}{K_{in}}} + \lambda \cdot f_i \cdot \rho \qquad (2\text{-}25)$$

In this equation, the different terms are in order from left to right: production from enzymatic systems, incorporation into proteins, dilution, transport from the surrounding medium and release from existing proteins by their degradation.

Also, X_i is the concentration of amino acid i, K_p is the K_m-value (Michaelis-Menten) of the enzyme producing amino acid i, Φ_i is the concentration of proteins transporting amino acid i into the cell from the medium with the K_m-value K_{in} and ρ is as defined in 2-22.

This equation is time consuming to solve numerically. However, the flows of amino acids from synthesis, tRNA charging and incorporation into proteins can be viewed as one single flow of metabolites. By putting the three flows equal, the calculations can be speeded up a lot. The different flows that are put equal in order to speed up the algorithm are the following:

$$\underset{1}{\frac{r\,\mathrm{f}_i}{Sum + \frac{\mathrm{f}_i\,(1 + \frac{K_m}{\mathrm{T}_{tot\,i} - \mathrm{T}_i})}{\mathrm{k}_{r\,i}}}} = \underset{2}{\frac{\mathrm{X}_i\,S_0\,k_2}{\frac{k_2}{k_0} + \frac{\mathrm{X}_i\,K_s}{\mathrm{T}_i}}} = \underset{3}{\frac{\mathrm{k}_i\,\mathrm{e}_i}{1 + \frac{\mathrm{X}_i}{K_p}} + \frac{\mathrm{f}_i\,\Phi_\mathrm{i}}{1 + \frac{\mathrm{X_i}}{\mathrm{K_{in}}}}} \qquad (2\text{-}26)$$

In these flows i is the index of the amino acid for which the flow is calculated. The first flow is the consumption of amino acids by incorporation into proteins by the ribosome. Here Sum is the sum of the time it takes for the ribosome to incorporate all amino acids but the i:th one. The other term of the denominator is the time it takes for the i:th amino acid, where K_m is the K_m-value of the ribosome, T_i is the concentration of *charged* tRNA cognate for amino acid i and $\mathrm{T}_{tot\,i}$ is the total concentration of tRNA cognate for amino acid i.

The second flow is the charging of tRNA, where S_0 is the total synthetase concentration, k_2 is the rate constant for amino acylation, k_0 is the rate constant

for binding of an amino acid to the synthetase and K_s is the dissociation constant for tRNA binding to the synthetase. The third flow is synthesis of amino acid i by enzyme e_i plus the uptake from the medium as defined in 2-25.

3 Implementation of the model

The model is at a basic level built up around differential equations for the concentrations of the different metabolites. The entire model is also deterministic with all advantages and drawbacks that come with it. In order not to get a 120-dimentional problem to solve in each iteration, I chose to make some simplifications of the calculations; namely viewing the flow of amino acids and mRNAs as being used directly into the translation without ever having to be calculated explicitly. This was justified by the theories presented in section 2.

3.1 Separation of time scales

To simplify the calculations and increase the speed of the program, I used the fact that some reactions finish faster than others. If the time step is chosen to successfully calculate the rates of some slow reactions, the time step will be large enough for fast reactions to reach equilibrium within a single iteration. I. e. for the purpose of calculating the rates of the slow reactions, the fast ones are considered to be in equilibrium. However, since the equilibrium is renewed in every time step, the concentrations will still have to be re-evaluated in each time step.

Since the flow of reactants in an equilibrated reaction is zero, one gets a closed expression for the concentrations of the reactants by setting the time derivative to zero in the differential equation for that reaction. The resulting expression is then treated with Maple to achieve a closed form for the required concentration. See Appendix C for the calculations.

3.2 Programs

The choice of programs used to model my system was MATLAB for the coding and Maple for calculation of the explicit expressions. In order for the program to run faster, I used a c++ compiler for MATLAB to achieve a stand-alone executable instead of a MATLAB function. The MATLAB-code is presented in Appendix A.

3.3 Algorithms

The model had to account not only for numerical solutions of differential equations, but also variable time delays, implicit calculations and variable time steps. Various strategies had to be used to implement these features in MATLAB-code. The program was built around finding a numerical solution to the system of differential equations in each iteration. Variable time delays for the emergence of active proteins as described in section 2.6 were introduced. These time delays were implemented using dynamic memory allocation and explicit expressions calculated from fast equilibria.

3.3.1 System of differential equations

Numerical solutions to the system of differential equations were implemented by a simple Runge-Kutta solver using the following algorithm:

$$\begin{aligned}
\vec{X}\Big|_{t_{n+1}} &= \frac{1}{6}\left(d\vec{X}_0 + 2d\vec{X}_1 + 2d\vec{X}_2 + d\vec{X}_3\right) \quad \text{where} \\
d\vec{X}_0 &= \frac{d}{dt}\,\vec{X}\Big|_{t_n} \\
d\vec{X}_1 &= \frac{d}{dt}\left(\vec{X}\Big|_{t_n} + \frac{1}{2}d\vec{X}_0\right) \\
d\vec{X}_2 &= \frac{d}{dt}\left(\vec{X}\Big|_{t_n} + \frac{1}{2}d\vec{X}_1\right) \\
d\vec{X}_3 &= \frac{d}{dt}\left(\vec{X}\Big|_{t_n} + \frac{1}{2}d\vec{X}_2\right)
\end{aligned} \tag{3-1}$$

In this algorithm, the vector $\vec{X}\Big|_{t_n}$ contains the values of all the non-fast equilibrating concentrations evaluated at time t_n. Those are the concentrations of biosynthetical enzymes, ribosomal proteins and ppGpp hydrolase. All other concentrations can either be computed explicitly from fast equilibria or achieved implicitly from a flow of reactants.

Since some of the concentrations calculated numerically vary a lot, I introduced a self-adjusting time step. I. e. if the rate of change is large, the time step will decrease. This will counter the loss of accuracy otherwise experienced.

3.3.2 Time delays

The time delays in gene expression were implemented by saving the concentrations of *initiated* translations of mRNAs in a matrix for each time step and protein synthesised. This leads to a dynamically allocated matrix on the following form:

$$\begin{bmatrix}
\vdots & \vdots & \vdots \\
[e_1]_{t_n} & \cdots & [e_a]_{t_n} \\
[e_1]_{t_{n+1}} & \cdots & [e_a]_{t_{n+1}} \\
\vdots & \vdots & \vdots
\end{bmatrix} \tag{3-2}$$

One element of the matrix is the concentration e_k of the k:th protein currently being synthesised. The translation of the mRNA for that protein was initiated at time t_i in the i:th time step.

In order to keep track of when a protein is finished and becomes active, I introduce a progress matrix with the same size as the matrix 3-2 on the following form:

$$\begin{bmatrix}
\vdots & \vdots & \vdots \\
\left[P^1_{t_n}\right]_T & \cdots & \left[P^a_{t_n}\right]_T \\
\left[P^1_{t_{n+1}}\right]_T & \cdots & \left[P^a_{t_n}\right]_T \\
\vdots & \vdots & \vdots
\end{bmatrix} \tag{3-3}$$

The elements of the progress matrix correspond to the same elements in the protein matrix. An element contain the total amount of amino acids, P, already incorporated into the corresponding enzyme at time T, where t_i is the time of initiation of translation for that specific protein.

For each time step the concentration of initiated proteins are saved in the matrix 3-2 and the progress in matrix 3-3 will be increased by the number v (the velocity of the ribosome). If the progress $P^k_{t_n}$ reach the total length of the k:th protein at some time T, the concentration of active protein k will be increased by the corresponding amount e_k from matrix 3-2. In order to prevent the same protein to be added several times, the progress indicator $P^k_{t_n}$ will be set to -1 to indicate that the protein has been released and is now active.

If 50 consecutive rows of the progress matrix have all elements being -1, those 50 rows of matrix 3-3 and matrix 3-2 will be removed from the memory to ensure that we have enough space to save later values. If the matrix isn't large enough to support all values, 50 new rows will be added at the bottom of both matrices.

3.4 Testing the model

When testing the model I compared the behaviour of my in silico cell during amino acid starvation with the experimental data from literature. In doing so I realised the importance of having the correct starting values. I had to do long simulations just to calibrate the system, before I even could induce the stringent response. It also became obvious that the system was very sensitive to changes in the concentrations. Unlike a living cell, a model of a cell does not begin with physiological values of the parameters. This means that I had to let the system adjust for several hours or even days just to get useful parameter values. These calibrations needed to be done every time I ran the program which made it a time consuming procedure to run the program. However, since the model is deterministic, it is possible to save the system and continue at a later time. This meant that I soon skipped the calibration and loaded the parameters at start-up.

While testing the model I soon discovered that the entire system seemed to work as anticipated. The ppGpp levels did indeed control the ribosome in a way such that the binding to the promoter for rRNA was strongly inhibited as the charging of tRNAs shrunk. Also, the production was shifted towards enzymatical systems producing amino acids during nutritional stress. This all seemed to be a correct description of the stringent response. However, some of the subsystems did not behave as expected. The production of ribosomes was very tricky to control and is by the time of writing not solved entirely. This is a bit of a disappointment, but will probably be solved in the future.

4 Results

The results from this work turned out to be both interesting and frustrating. They showed, however, that it is indeed possible to form closed expressions for many of the regulatory systems in the cell, which speeds up the numerical computations significantly.

The graphs in this section all arise from plotting concentrations of metabolites against simulation time. The plotting begins on a system that is in a state of exponential growth. This means that the system already calibrated for a long time which is not shown in the graphs. The first thousand seconds in the simulation are sometimes used to set the values of the matrices involved and therfore these seconds might not show correct values. The colours in the graphs each denote the concentration of one amino acid or tRNA carrying or enzymes producing that amino acid.

4.1 ppGpp and the ribosome

Preliminary results from the model were promising. They showed that the proposed regulation of ribosome synthesis by ppGpp is a plausible mechanism for the stringent response and growth optimisation. This is illustrated in Figure 3. Here, the level of ppGpp, the concentration of ribosomes and the level of amino acids available to the ribosome all show the expected behaviour when the import of ten out of twenty amino acids into the cell is reduced. The nutritional downshift causes the level of ppGpp to rise rapidly, the level of ribosomes to shrink and the levels of the different amino acids to either increase or to be drastically reduced, depending on which one of them that is rate limiting for protein synthesis. That the concentrations of ppGpp, ribosomes and amino acids change, as indicated by the model, when amino acid starvation is introduced, is well documented in numerous experimental studies (Ehrenberg, M., Bremer, H. and Dennis, P.; Control of ribosomal RNA synthesis in *Escherichia coli*; manuscript in preparation).

4.2 tRNA and amino acid biosynthetic systems

During the simulations the relative levels of charged tRNA altered between roughly zero and one. It was also only one tRNA at a time that had a charge ratio of zero as shown in Figure 4. This graph arose by plotting the different ratios of charging for the twenty tRNAs in a cell in exponential growth against time. As the model assumes that only one amino acid is limiting the system at any time, this is expected. It is also not the same amino acid that is limiting protein synthesis; again an expected result.

The behaviour of only one amino acid limiting the growth at a time also affect the concentrations of the amino acid producing enzymatic systems. This can be seen by comparing the concentrations of the enzymes to the charge ratios of tRNA. By plotting the concentrations of the enzymes for the same time period as the charge ratios, as seen in Figures 4 and 5, one can see that an amino

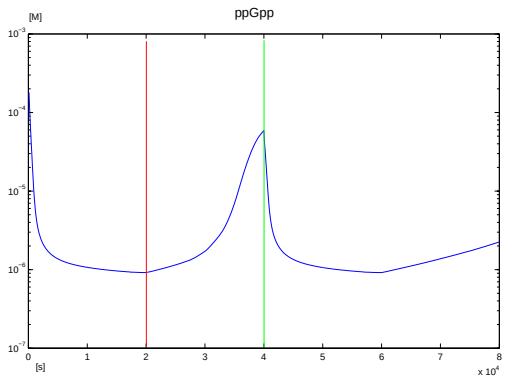


Figure 3: A graph showing the concentration of ppGpp on the y-axis and simulation time in seconds on the x-axis. At time $2 \cdot 10^4$ seconds a nutritional downshift is introduced. At time $4 \cdot 10^4$ seconds the nutritional environment is shifted back again.

acid limiting the growth also increases the production of enzymes producing the same limiting amino acid. This shows that the production of an enzymatic system is switched on correctly as the availability of the corresponding amino acid becomes low. When the amino acid limitation is removed, the production of the corresponding enzyme decreases and its concentration decreases by dilution and degradation. This explains the "quick rise, slow fall"-features of the curves in Figure 5. The concentration of an enzyme increases rapidly until its corresponding amino acid is not limiting anymore. Then the concentration of that enzyme decreases until its amino acid is limiting again, if that ever occurs again.

One can simulate the amino acid starvation by setting the rate of transport from the surrounding medium for one or many amino acids to zero. When introducing such a nutritional shift by removing the external supply of ten of the amino acids all at once, one can see that the rate of production of the enzymes for the ten amino acids no longer in the medium is greatly enhanced to cover for the loss of externally supplied amino acids. A graph of the concentrations of the enzymes plotted against time during this starvation can be seen in Figure 6. This graph also contain the behaviour of the concentrations of the enzymes when the starvation is lifted. Further it shows what happens if all amino acids are removed from the medium.

4.3 Quasi-steady states

Some of the reactions involved in the system were solved analytically and introduced in closed form in the calculations to improve computational speed. These cumbersome calculations are shown in Appendix C. The first subsystem I used this strategy for was the system 2-13, and this led to a rather nice expression for the ppGpp producing complex ART only depending on known concentrations. In the expression below, A_0 is the level of free A sites on ribosomes, R_0 is the (constant) level of total PSI in the cell and T_0 is the (constant) level of total

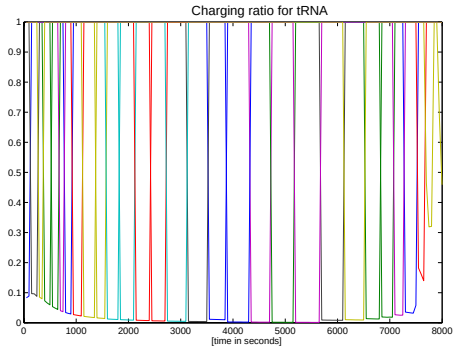


Figure 4: A graph showing the charging ratios for the different tRNAs during exponential growth on the y-axis and simulation time in seconds on the x-axis. The tRNAs' charging ratios alter between zero for the limiting amino acid, and one for those who are not. I. e. only one tRNA has a charging ratio of zero at a time, and that tRNA is the one carrying the limiting amino acid. Its charging ratio stays close to zero until another amino acid becomes limiting.

tRNA in the cell.

$$ART = \frac{2.5 \cdot 10^{-11}}{A0} \cdot \left(-10^5\, A_0 - 10^5\, T_0 - 1 + \Gamma\right) \cdot \left(-10^5\, R_0 - 10^5\, A_0 - 1 + \Delta\right)$$

where

$$\Gamma = \sqrt{10^{10}\, A_0^2 - 2 \cdot 10^{10}\, A_0\, T_0 + 2 \cdot 10^5\, A_0 + 10^{10}\, T_0^2 + 2 \cdot 10^5\ T_0 + 1}$$

$$\Delta = \sqrt{10^{10}\, R_0^2 - 2 \cdot 10^{10}\, R_0\ A_0 + 2 \cdot 10^5\, R_0 + 10^{10}\, {A_0}^2 + 2 \cdot 10^5\, A_0 + 1} \tag{4-1}$$

Another quasi-steady state that could be written in closed form was the level of ppGpp in the cell. This was the most complex calculation in the project. However, the behaviour of the concentration of ppGpp varied according to expectation; rising or falling depending on the relation between supply and demand for amino acids in the cell.

The more details put into the model to simulate the stringent response of a cell, the slower the calculations became. This led me to do a lot of alterations to the model by introducing the flow of metabolites as shown in section 2.7, which replaced twenty dimensions of the differential equation system with just one explicit calculation in each iteration. The computational time for one simulation decreased from thousands of years to hours. The calculations can be seen in Appendix D and the closed expression for the flow in 2-26.

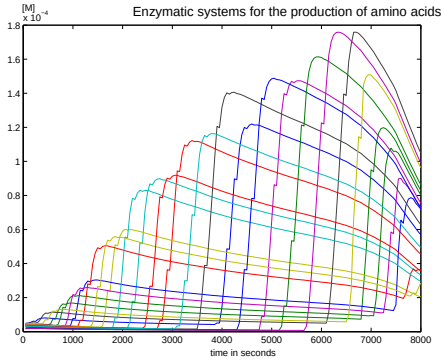


Figure 5: A graph showing the concentration of enzymes on the y-axis and simulation time in seconds on the x-axis. The concentration of enzymes producing amino acids increase as the charging of the corresponding tRNA becomes low. Compare to Figure 4. The time delay caused by the actual production by ribosomes makes the enzymes slowly increase to a level where the production is turned to another amino acid system. From this point the enzymes are starting to decrease due to dilution and degradation.

5 Discussion

5.1 General comments

The main question of this work is of course whether the model satisfactorily describes the biological system. Naturally, it is not a perfect model, but can it bring understanding of how the cell works? I think it can. Maybe not by verifying actual results, but by telling us that some things considered unnecessary to include, might indeed prove vital to the cell. E. g. when introducing variable time spans for completing a protein, one directly sees that the controllability of the system decreases; the time lag makes the system much more sensitive to programming and parameter errors. The discouraging conclusion from this is that the time and money it takes to complete a model increases tremendously with its complexity.

A thing often referred to when talking about computations is More's law. This law states that the computational power doubles in 18 months. This could of course in some respects help programming projects like this one, but in a deeper sense it has nothing to do with the real problem. Without a correct setup of the system, the calculations wouldn't be correct even if the computer worked with infinite speed. This is due to the fact that if you don't know what you are measuring, it can't be modelled. This was one of the points I had to think about. Do I really model the correct things? Is something perhaps not what I think it is? Those questions are hard to cope with, even if there are clarifying experiments that can be done. Nevertheless, those questions are necessary in order to achieve results based on facts and not mere hopes.

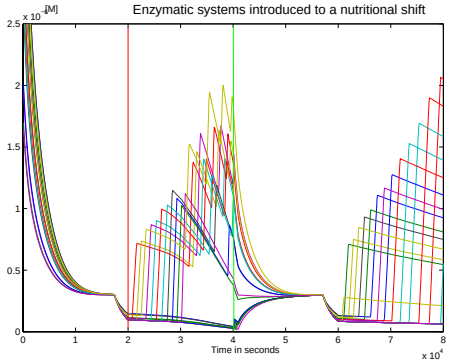


Figure 6: A graph showing the concentration of enzymes on the y-axis and simulation time in seconds on the x-axis. The alternating sequence of the enzymes is interrupted by the stringent response at time $2 \cdot 10^4$ seconds. That causes the enzymes producing the limiting amino acids to increase very fast as those amino acids no longer exists in the medium. At time $4 \cdot 10^4$ seconds, the removed amino acids are put back in the medium. At time $6 \cdot 10^4$ seconds, *all* amino acids are removed from the medium. This causes massive lack of amino acids, and the concentrations of all enzymes are thus increased enormously.

When modelling something as vast as an entire cell it is essential to make simplifications. The ones introduced here are all based on theoretical knowledge and real experiments. The main problem with this is that many of the subsystems involved are poorly understood. This will of course result in some kind of strange deduction game, where you hold some things for facts and have to deduce and try other things. The fact that many of the interactions between ppGpp and other molecules are completely unknown increased the uncertainty in my model. Actually, I can only say that I have made a simulation of a probable system that could bear some resemblance to a living cell.

5.2 The model

The model consists of two major parts: The assembly of amino acids into all kinds of proteins and the regulation of the same production. These two aspects are centred on the ribosome for synthesis and the effector molecule ppGpp for its control.

5.2.1 ppGpp

The model for synthesis and degradation of ppGpp is indeed a very appealing one. Its different ways of production and use (see section 2) are showing great resemblance to what today is believed to be the correct pathway. The ribosome-tRNA^c-PSI complex model does indeed seem very plausible (Ehrenberg, M., Bremer, H. and Dennis, P.; Control of ribosomal RNA synthesis in *Escherichia coli*; manuscript in preparation). Even though some of the involved steps, e.

g. the *spoT* path producing both PSII and a ppGpp hydrolase, remains to be clarified, the model does not contradict any of today's knowledge. Since this is the actual control part of the system it seems that getting a correct model for this is what essentially makes for a correct stringent response.

5.2.2 The ribosome

When modelling the production of proteins by ribosomes, one has to take into account that the ribosome consists of rRNA and ribosomal proteins. The latter is produced by the ribosome, and the first is produced by RNA polymerase, which itself is a protein made by the ribosome. This will make the system itself contain somewhat of a feed forward loop: the more ribosomes we have, the more we can produce. Add to this that the ribosomes indirectly produce the amino acids that are used and you have a worthy task for the ppGpp to regulate. Since the ribosome is dependent on itself in many ways, it becomes hard to predict what would happen when I changed something in the program. One of the result of this can be seen in Figure 7. These graphs show the concentrations of ppGpp

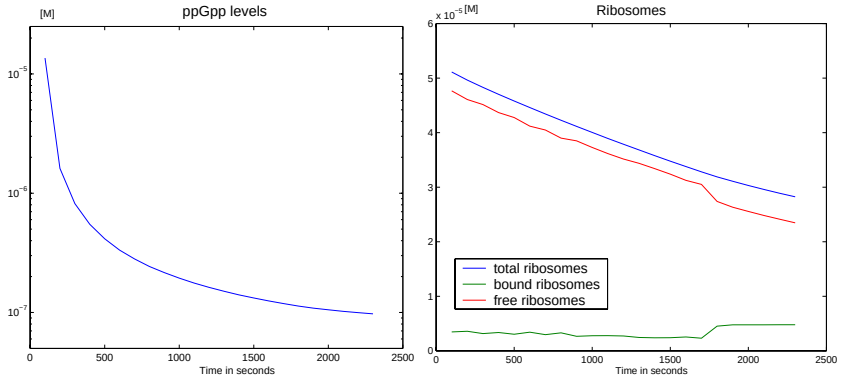


Figure 7: Two graphs showing the concentration of ribosomes to the right and the concentration of ppGpp to the left. On the x-axis is simulation time in seconds.

and ribosome taken from the point at which the starvation is over and ppGpp is no longer needed to suppress the production of ribosomes. These graphs show the same period of time as the right part of Figure 3 where the concentration of ppGpp is decreasing again after the stringent response. Since this should lead to an increase of the concentration of ribosomes, it is clearly an unwanted result that the concentration of ribosomes decreases without ever being able to recover, even if ppGpp levels drop. This might be a consequence of the complexity of the model, leading to unexpected difficulties in the implementation. It might also be that the control mechanism suggested for the ribosome isn't powerful enough to cope with the large variations in the concentrations caused by the time delays in the model. A third possibility is that there might be some error in the programming.

5.3 Outlook

If more time and effort would be put into this project I am sure that the bugs would be hunted down and the quirks of the system removed. The enzymes, amino acids, ppGpp and RNAs are all working fine and since it is virtually only the ribosome that does something strange, I am sure that the problem will be solved and the model proved correct.

5.3.1 Numericals

It would also be possible to further improve the speed of the calculations by using another method for numerical integration. The Runge-Kutta method used isn't the fastest. Furthermore, the speed of the computations could be improved a lot by being implemented in C++ or Fortran instead of MATLAB, since MATLAB isn't good at dealing with dynamic memory allocation.

6 Acknowledgments

Jag vill tacka min handledare **Måns Ehrenberg** för att han tog sig an mig och visade mig vetenskapens sanna skönhet. Vidare vill jag tacka **Johan Elf** för ovärderlig hjälp och lärdom. Jag vill också tacka min examinator **Kurt Nordström**. Vidare vill jag tacka min vän **Andreas**, utan honom hade jag aldrig fått tummen ur. Tack även till mina opponenter Anders Bjurström och Kristoffer Forslund. Till sist vill jag tacka Donald E. Knuth för att han hittade på LaTeX 2_ε.

7 References

[1] H. Bremer, P. Dennis, and M. Ehrenberg. Free RNA polymerase and modeling global transcription in *Escherichia coli*. *Biochimie.*, 85:597–609, 2003.

[2] N. A. Campbell. *Biology*. Benjamin/Cummings Publishing, 4th edition, 1996.

[3] M. Cashel, D. R. Gentry, V. J. Hernandez, and D. Vinella. The stringent response. In Neidhardt F.C. et al. eds, editor, *Escherichia coli and Salmonella typhimurium: Cellular and Molecular Biology, 2nd edition*, volume 1, chapter 92, pages 1458–1496. American Society for Microbiology, Washington D. C.

[4] J. Elf, O. Berg, and M. Ehrenberg. Comparison of repressor and transcriptional attenuator systems for control of amino acid biosynthetic operons. *J. Mol. Biol.*, 313:941–954, 2001.

[5] M. Ibba and D. Söll. Aminoacyl-tRNA synthesis. *Annu. Rev. Biochem.*, 69:617–650, 2000.

[6] K. D. Murray and H. Bremer. Control of spoT-dependent ppGpp synthesis and degradation in *Escherichia coli*. *J. Mol. Biol.*, 259:41–57, 1996.

[7] F. C. Neidhardt, J. L. Ingraham, and M. Schaechter. *Physiology of the Bacterial Cell. A Molecular Approach.* Sunderland, Massachusetts, USA: Sinauer Associates, Inc., 1990.

[8] P. O'Farrell. The suppression of defective translation by ppGpp and its role in the stringent response. *Cell*, 14:545–557, 1978.

[9] E. Pärt-Enander and A. Sjöberg. *Användarhandledning för MATLAB5*. Uppsala Universitet, Uppsala, 1998.

[10] T. M. Wendrich, G. Blaha, D. N. Wilson, M. A. Marahiel, and K. H. Nierhaus. Dissection of the mechanism for the stringent factor *relA*. *Mol Cell.*, 10:779–788, 2002.

8 Appendices

Appendix A

The main program containing everything except the actual time step:

```
%function Modell_test
tic;
clear;
%I load the starting values from a file where they have been calibrated. The starting values for that calibration are given as comments below.
load start2;
% aaa = 20;
% X = X(1:aaa); e = e(1:aaa); mRNA = mRNA(1:aaa); Ttot = Ttot(1:aaa); T = T(1:aaa); alphax = alphax(1:aaa);
% tidde = tidde(:,1:aaa);
%
% %---------givna-----------
% S0 = 1.3e-6;
% k2 = 100;
% k0 = 1e6;
% Ks = 1e-6;
% Kp = 5e-5;%3; %M 4n 3e 4hn 4ho 3he
% rho = 2; % M
% f = .001 .* [ 85 70 40 46 21 56 85 59 63 59 47 60 58 36 22 16 46 38 15 78 ]';
% % %              A  V  F  P  M  I  L  D  E  K  R  S  T  Y  H  C  N  Q  W  G
% L    = [ 6000  6000  2000  1000  1000 ];
% c    = [ 85  85  45  45  1 ];
% for i = 1:aaa
%     faktor(i) = 5e-4;
%     kr(i) = 15; % 1/sec
%     k(i) = 2 * f(i); % 1/se
%     Ttot(i) = 1e-5; %M
% end
% if aaa ~= 20
%     f = f(1:aaa);
%     f = f ./ sum(f);
% end
% Km = 5e-6;%7; %aa-rib interakt  5n 7e 6hn 7ho 6he
% krna = 5e-9; %M/sec
% lambda = 1/100; %per sec
% Kr = 1e-5; %M
% Kx = 1e-6; %M
% %-----------------------
%
% %------------startgissningar---------------
% for i = 1:aaa
%  e(i) = 1e-5;
%  X(i) = 1e-4;
%  mRNA(i) = 1e-7;
%  alphax(i) = 0.05;
% end
% T = 0.1 * Ttot;
% r = 2.5e-5; %M
% ppGpp = 5e-7;
% RNAP = 1e-5;
% RNAPdel = 5e-6;
% RNAPbundet = RNAP - RNAPfri;
% totRelA = 1e-4; %M fetgissning
% slaskpool = 0.8; %M cellkeeping i aa
% hydrospoT = 0; %M fetgissning
% %------------------------------------------
%-------------värdesättning-------------
Totaltid = 80000; % sekunder
lol = 100;
mem = 10000;
dtstart = 1; %sekund
plott1  = zeros(mem,1);
plott2  = zeros(mem,aaa);
plott3  = zeros(mem,aaa);
plott4  = zeros(mem,1);
plott5  = zeros(mem,aaa);
plott6  = zeros(mem,aaa);
plott7  = zeros(mem,1);
plott8  = zeros(mem,3);
plott9  = zeros(mem,aaa);
plott10 = zeros(mem,1);
timeaxis = zeros(mem,1);
v = 0; tau = 0; summa = 0; tmp = 0;
count = 0; current_time = 0;
procent = 0; time_left = 0; timmar = 0; minuter = 0; sekunder = 0;
decay = 5e-5; %För mycket nedbrytning innan faktor 100

%---tidsfördröjning---
% transtermer = zeros(1000,5);
% etranstermer = zeros(1000,aaa+1);
% transterm = [ 0 0 0 0 0 ];
% etransterm = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
% vilken = [ 1 1 1 1 1 ];
% helt_steg = [ 0 0 0 0 0 ];
% chunk = 64;
% eflow = zeros(chunk, aaa+2); %aaa+1 för spoT
% progress = zeros(chunk, aaa+2);
% einit = zeros(1, aaa+2);
```

```
% pekare = ones(1, aaa+2);
% steg = 0;
% % %---------------------
% tol = 0.1; %hur stor får en ökning vara
% koll = 0; tRNAfel = 0; Tkoll = 0; T_gammal = T;
%----------------------------------------

%--------Ändringar vid inladdning------------
% eflow = [ eflow, zeros(size(eflow,1),1) ];
% progress = [ progress, zeros(size(progress,1),1) ];
% einit = [ einit, 0 ];
% pekare = [ pekare, 1826 ];
% etranstermer = [ etranstermer, zeros(size(etranstermer,1),1) ];
% etransterm = [ etransterm, 0 ];
% e = [ e, r ];
% alphax = alphax(1:aaa);
% mRNA = [ mRNA, 0 ];
% ppGpp = 5e-7;
% etranstermer(:,21) = etranstermer(:,21).*0.01;
% q = [2 2 2 2 2 2 2 2];
% transtermer = transtermer(1:10,:);
% etranstermer = etranstermer(1:10,:);
% vilken = [1 1 1 1 1];
% helt_steg = [1 1 1 1 1];
r = e(aaa+2);
rRNA = .5*e(aaa+2);
RNAPfri = 1e-6;
for i = 1:aaa
    kr(i) = 20; % Ändra maxhastigheten på ribosomen
end
eflow =abs(eflow);
%-----------------------------------------------

fig = figure(1);
set(fig,'Position',[1 1 1152 796]);

% loopa här
while current_time < Totaltid
    dt = dtstart;
    for_stort_dt = 1;

    %-----------ändra miljön---------
    if (current_time > 1000)
        faktor(10:20) = 0;
    end
    if (current_time > 40000)
        faktor(1:20) = 5e-4;
    end
    if (current_time > 60000)
        faktor(1:20) = 0;
    end
    %----------------------------------

    %---beräkna värden-------
    v = 0;
    V = 0;
tau = 0;
summa = 0;
    %--Beräkna uppdelningen av ribbar--
    %Om den här djävla slaskpoolen inte vill hålla sig på mattan sätter jag den till ett bättre värde
    boundrib = (slaskpool + 1000*sum(e(1:aaa+1))+10000*e(aaa+2))/(1000*sum(e(1:aaa+1))+10000*e(aaa+2)) * (sum(sum(eflow(:,1:aaa+1)))+50*sum(eflow(:,aaa+2)));
    %Ity 1:1-förhållande mellan ribosomer och enzymer 200 aa i 50 st ribprot
    if boundrib + eps > r
        boundrib = r - eps;
    end
    Rib_fri = r - boundrib;
%----------------------------------
    %----------------Beräkna v, tau, mu, rho---------------
    V = ( k.*e(1:20) + f'.*faktor + decay * f' * ( sum(e(1:aaa+1))*1000 + e(aaa+2)*10000 + RNAPfri*1000) ) ./ Rib_fri ./f';
[v, index] = min(min(V, kr));
    for i = 1:aaa
        tau(i) = 1/kr(i);
    end
    tau(index) = 1/(v*f(index)) - (1-f(index)) / (f(index)*kr(index));
    rho = slaskpool + 1000 * sum(e) + 10000 * e(aaa+2) + 1000 * RNAPfri; % M
    mu = Rib_fri * v / rho;
    %-------------------------------------------------------------
    %Ersätter faktor(i) med ( faktor(i)+ decay * f' * ( sum(e(1:aaa+1))*1000 + e(aaa+2)*10000 + RNAPfri*1000) ) ) för bättre X-beräkningar
    %--------X, T, T3--------
    for i = 1:aaa
        alphax(i) = 1 - exp( -(tau(i)-1/kr(i)) );
        T3(i) = Ttot(i);
        T(i) = Ttot(i) - T3(i);
        X(i) = -Kp*(Rib_fri*v*f(i)-k(i)*e(i)-f(i)*( faktor(i) + decay * ( sum(e(1:aaa+1))*1000 + e(aaa+2)*10000 + RNAPfri*1000 ) ) )/(Rib_fri*v*f(i));
end
    if tau(index) > 1/kr(index) + eps
        T3(index) = Km / ( tau(index)*kr(index) - 0.5 );   %Bytte 1 mot 0.5 för att få bättre siffror
        T(index) = Ttot(index) - T3(index);
        X(index) = -k2*T(index)*(k(index)*e(index)+f(index)*( faktor(index) + ...
        decay * ( sum(e(1:aaa+1))*1000 + e(aaa+2)*10000 + RNAPfri*1000 ) ) )/k0/(k(index)*e(index)*Ks + ...
f(index)*( faktor(index) + decay * ( sum(e(1:aaa+1))*1000 + e(aaa+2)*10000 + RNAPfri*1000 ) )*Ks-S0*k2*T(index));
        %1/2/(-f(i)*faktor(i)*Ks*k0+S0*k2*k0*T(i))*(k(i)*e(i)*Kp*Ks*k0+f(i)*faktor(i)*Kp*Ks*k0+f(i)*faktor(i)*k2*T(i)-S0*k2*k0*T(i)*Kp+...
sqrt(k(i)^2*e(i)^2*Kp^2*Ks^2*k0^2+2*k(i)*e(i)*Kp^2*Ks^2*k0^2*f(i)*faktor(i)-2*k(i)*e(i)*Kp*Ks*k0*f(i)*faktor(i)*k2*T(i)-...
2*k(i)*e(i)*Kp^2*Ks*k0^2*S0*k2*T(i)+f(i)^2*faktor(i)^2*Kp^2*Ks^2*k0^2-2*f(i)^2*faktor(i)^2*Kp*Ks*k0*k2*T(i)-...
2*f(i)*faktor(i)*Kp^2*Ks*k0^2*S0*k2*T(i)+f(i)^2*faktor(i)^2*k2^2*T(i)^2+2*f(i)*faktor(i)*k2^2*T(i)^2*S0*k0*Kp+S0^2*k2^2*k0^2*T(i)^2*Kp^2+...
4*S0*k2^2*k0*T(i)^2*k(i)*e(i)*Kp))
    end

    %----------------------
```

```
    tmp = sum(alphax);
if tmp <= 0
        alphax = ones(1,aaa);
        tmp = sum(alphax);
end
    if tmp > 1
        alphax = alphax / tmp;
end
    %-------------------------
%     boundrib = (slaskpool + sum(e)*1000)/(1000*sum(e(1:aaa+1))+10000*e(aaa+2)) * (sum(sum(eflow(:,1:aaa+1)))+50*sum(eflow(:,aaa+2)));
%Ity 1:1-förhållande mellan ribosomer och enzymer 200 aa i slaskenzym
    friAsiterib = boundrib - ( .5*( boundrib + sum(f'.*T3) + 1e-7 - sqrt( (boundrib+sum(f'.*T3)+1e-7).^2 - 4*boundrib*sum(f'.*T3)) ) );
    %Hur mkt translaterande ribbar med fritt A site
    T0 = sum(Ttot-T3); A0 = friAsiterib; R0 = totRelA;  %Maplekonvertering
    ART = (R0+T0+1e-5)/2 + sqrt((R0+T0+1e-6)^2/4-T0*R0);
    %.2500000000e-10*(100000.*A0+1.+100000.*T0-1.*sqrt(.1000000000e11*A0^2+200000.*A0-.2000000000e11*A0*T0+1.+200000.*T0+...
    .1000000000e11*T0^2))*(100000.*R0+1.+100000.*A0-1.*sqrt(.1000000000e11*R0^2+200000.*R0-.2000000000e11*R0*A0+1.+200000.*A0+...
    .1000000000e11*A0^2))/A0;
%     AR = .2500000000e-10*(100000.*R0+1.+100000.*A0-1.*sqrt(.1000000000e11*R0^2+200000.*R0-.2000000000e11*R0*A0+1.+200000.*A0+...
    .1000000000e11*A0^2))*(-100000.*T0+100000.*A0-1.+sqrt(.1000000000e11*A0^2+200000.*A0-.2000000000e11*A0*T0+1.+200000.*T0+...
    .1000000000e11*T0^2))/A0;
    RelAterm = ART;
    %--------------ta ett steg ----------------------
    while ( for_stort_dt > 0)
        [ ppGpp, spoTdel, relAdel, hydrodel, de0, dhydrospoT0, mRNA, drRNA0 ] = Derivata( r,  rRNA,            tau,  e,         k, Kp, v, f, ...
mu, Kr, alphax, lambda, ppGpp, aaa, faktor, RelAterm,  hydrospoT,                 Ttot, kr, Km, decay, boundrib );
        [ ppGpp, spoTdel, relAdel, hydrodel, de1, dhydrospoT1, mRNA, drRNA1 ] = Derivata( r, (rRNA+.5*drRNA0), tau, (e+.5*de0), k, Kp, v, f, ...
mu, Kr, alphax, lambda, ppGpp, aaa, faktor, RelAterm, (hydrospoT+.5*dhydrospoT0), Ttot, kr, Km, decay, boundrib );
        [ ppGpp, spoTdel, relAdel, hydrodel, de2, dhydrospoT2, mRNA, drRNA2 ] = Derivata( r, (rRNA+.5*drRNA1), tau, (e+.5*de1), k, Kp, v, f, ...
mu, Kr, alphax, lambda, ppGpp, aaa, faktor, RelAterm, (hydrospoT+.5*dhydrospoT1), Ttot, kr, Km, decay, boundrib );
        [ ppGpp, spoTdel, relAdel, hydrodel, de3, dhydrospoT3, mRNA, drRNA3 ] = Derivata( r, (rRNA+.5*drRNA2), tau, (e+.5*de2), k, Kp, v, f, ...
mu, Kr, alphax, lambda, ppGpp, aaa, faktor, RelAterm, (hydrospoT+.5*dhydrospoT2), Ttot, kr, Km, decay, boundrib );
        for_stort_dt = (dt*RK(dhydrospoT0,dhydrospoT1,dhydrospoT2,dhydrospoT3) > (e(aaa+1)-hydrospoT)) + ( ppGpp < 0 ) + ...
(dt*RK(drRNA0,drRNA1,drRNA2,drRNA3) > .1 * rRNA);
        if ( for_stort_dt > 0 )
            dt = dt * .1;
        end
    end
    hydrospoT = hydrospoT + dt*RK( dhydrospoT0, dhydrospoT1, dhydrospoT2, dhydrospoT3 );
    rRNA = rRNA + dt*RK( drRNA0, drRNA1, drRNA2, drRNA3 );
    %------------Enzymflöde----------------
    einit = dt*RK( de0, de1, de2, de3 );
    %-----Kolla att det inte initierats mer än möjligt---------
    if boundrib + sum(einit(1:aaa+1)) + 50*einit(aaa+2) + eps > r
        einit = einit ./ (sum(einit(1:aaa+1)) + 50*einit(aaa+2) + eps) * Rib_fri;
    end
    %-------------------------------------------------------
    %---------Allokera mer om nödvändigt----------------
    if max(pekare)+1 > size(progress,1)
        eflow = [eflow; zeros(chunk,aaa+2)];
        progress = [progress; zeros(chunk,aaa+2)];
    end
    %-------------------------------------------------
    % Bind inte upp ribbarna nu utan efter hela tidssteget
    %----------Sätta in i sparmatris--------------
    for i = 1:aaa+2
        eflow(pekare(i), i) = einit(i);
        progress(pekare(i), i) = 0;
        pekare(i) = pekare(i) + 1;
    end
    %-----------------------------------------------
    %----------Propagera ribosomen och ta ut-------
    for i = 1:aaa+2
        for j = 1:pekare(i)
            progress(j, i) = progress(j, i) + dt * v;
            if progress(j, i) >= 1000 | ( progress(j, i) >= 200 & i == aaa+2) %För att ribbarna är 50 * 200;
                if i == aaa+1
                    Sht = v.^5 / ( 1e4 + v.^5);
                    hydrospoT = hydrospoT + eflow(j, aaa+1) * Sht;
                end
                e(i) = e(i) + eflow(j, i);          %Det tillverkade enzymet
                eflow(j, i) = 0;
                progress(j, i) = -1e9;
            end
        end
    end
    %-------Avallokera-----------
    for i = 1:aaa+2             %Här flyttar vi barra upp siffror i drift
        if max(progress(1:chunk,i)) < 0
            eflow(:,i) = [eflow(chunk+1:size(eflow,1),i);zeros(chunk,1)];
            progress(:,i) = [progress(chunk+1:size(progress,1),i);zeros(chunk,1)];
            pekare(i) = pekare(i) - chunk;
        end
    end
    if size(eflow,1) > max(pekare)+chunk            %Här tas minnet av dem bort.
        eflow = eflow(1:size(eflow,1)-chunk,:);
        progress = progress(1:size(eflow,1),:);
    end
    %-------------------------------
    %-----------------------------------------
    hydrospoT = hydrospoT - dt * mu * hydrospoT;
    e = e - dt * ( mu + decay ) * e;                        %Utspädningen och sönderfallet
    %---------ribbar-------------------
    r = min(rRNA, e(aaa+2)); %Beräkna totala ribbar
    %-----------------------------------
    count = count + 1;                                  % flytta räknaren
    current_time = current_time + dt;                   % beräkna tiden
%--------------tog ett steg------------------
```

```
    %--------plottvärden-----------
    if mod(count,lol) == 0
        timeaxis(count/lol) = current_time;
        plott1(count/lol,1) = RNAP; plott1(count/lol,2) = RNAPfri;
        plott2(count/lol,:) = X;
        plott3(count/lol,:) = mRNA(1:aaa);
        plott4(count/lol) = v;
        plott5(count/lol,:) = e(1:aaa);
        plott6(count/lol,:) = alphax(1:aaa);
        plott7(count/lol,1) = r; plott7(count/lol,2) = boundrib; plott7(count/lol,3) = Rib_fri;
        plott8(count/lol,1) = ppGpp; %plott8(count/lol,2) = spoTdel; plott8(count/lol,3) = relAdel; plott8(count/lol,4) = -hydrodel;
        plott9(count/lol,:) = T3./Ttot;
        plott10(count/lol) = mu;
        %----tid kvar----
        procent = floor(current_time/Totaltid*10000)/100;
        time_left = (toc/current_time .* (Totaltid - current_time))/3600;
        timmar = floor(time_left);
        minuter = floor((time_left-timmar) .* 60);
        sekunder = floor((time_left-timmar-minuter./60) .* 3600);
        [num2str(timmar),' h ',num2str(minuter),' m ',num2str(sekunder),' s kvar ',num2str(procent),'procent klart med tidssteg ',num2str(dt)]
        subplot(3,3,1);
%            if mod(count,2*lol) == 0
%  semilogy(timeaxis(1:floor(count/lol)), plott8(1:floor(count/lol),:));
%            else
           plot(timeaxis(1:floor(count/lol)), plott8(1:floor(count/lol),:));
%            end
        title(['                        ppGpp ',num2str(ppGpp),' Hydro = ',num2str(hydrodel)]);
        ylabel(['Växt = -',num2str(mu * ppGpp),' dppGpp = ',num2str(spoTdel + relAdel - hydrodel - mu * ppGpp)]);
        xlabel(['RelA = ',num2str(relAdel),' spoT = ',num2str(spoTdel)]);
%  legend('ppGpp','spoT','relA','hydrodel',3);
subplot(3,3,2);
plot(timeaxis(1:floor(count/lol)), plott9(1:floor(count/lol),:));
title(['tRNA-komplex Minsta T = ',num2str(min(T))]);
        xlabel(['Laddning = ',num2str(min(T3./Ttot)),' och ',num2str(max(T3./Ttot))]);
subplot(3,3,3);
        plot(timeaxis(1:floor(count/lol)), plott2(1:floor(count/lol),:));
        title(['Aminosyror, Minsta X = ',num2str(min(X))]);
        xlabel(['friAsiterib = ',num2str(friAsiterib), ' ART = ',num2str(ART)]);
        subplot(3,3,4);
        plot(timeaxis(1:floor(count/lol)), plott7(1:floor(count/lol),:));
        title(['Ribbar']);
        xlabel(['Prot: ',num2str(e(aaa+2)), ' rRNA: ',num2str(rRNA), ' Fria: ',num2str(Rib_fri), ' Tot: ',num2str(r)]);
        legend('Rtot','Rbundet','Rfritt',3);
subplot(3,3,5);
        plot(timeaxis(1:floor(count/lol)), plott4(1:floor(count/lol)),timeaxis(1:floor(count/lol)), 1e4 .* plott10(1:floor(count/lol)),'r');
        title(['v & mu*1e4']);
        xlabel(['V = ',num2str(min(V)),' v = ',num2str(v),' mu = ',num2str(mu)]);
subplot(3,3,6);
        plot(timeaxis(1:floor(count/lol)), plott5(1:floor(count/lol),:));
title(['e']);
        subplot(3,3,7);
        plot(timeaxis(1:floor(count/lol)), plott1(1:floor(count/lol),:));
        title([' totalt: ',num2str(RNAP),' fritt: ',num2str(RNAPfri)]);
        legend('RNAPtot','RNAPfritt',3);
        xlabel(['totalt: ',num2str(RNAP),' fritt: ',num2str(RNAPfri)]);
        subplot(3,3,8);
        plot(timeaxis(1:floor(count/lol)), plott3(1:floor(count/lol),:));
        title(['mRNA ',num2str(mRNA(aaa+1))]);
        xlabel([num2str(current_time),' s ']);
subplot(3,3,9);
        plot(timeaxis(1:floor(count/lol)), plott6(1:floor(count/lol),:));
        title(['alpha']);
        xlabel([num2str(procent),'procent klart med tidssteg ',num2str(dt)]);
        drawnow;
%         if mod(count,1000) == 0
%             save tmp;
%         end
        %---------------
    end
    %----------------------------
end


%--------fixa strängar---------
timmar = floor(current_time/3600);
minuter = floor((current_time/3600-timmar) .* 60);
sekunder = floor((current_time/3600-timmar-minuter./60) .* 3600);
Totaltid = [num2str(timmar),' h ',num2str(minuter),' m ',num2str(sekunder),' s i totaltid'];
%----------------------------

%-------plott------------
figure(1);
set(1,'Position',[1 1 1152 796]);
subplot(2,1,1);
plot(timeaxis(1:floor(count/lol)), plott1(1:floor(count/lol),:));
title(['RNAP , dt = ', num2str(dt)]);
legend('RNAPtot','RNAPfritt',3);
xlabel(Totaltid);
subplot(2,1,2);
plot(timeaxis(1:floor(count/lol)), plott2(1:floor(count/lol),:));
title(['Aminosyror , dt = ', num2str(dt)]);
xlabel(Totaltid);
% legend('A','V','F','P','M','I','L','D','E','K','R','S','T','Y','H','C','N','Q','W','G',-1)
figure(2);
set(2,'Position',[1 1 1152 796]);
subplot(2,1,1);
plot(timeaxis(1:floor(count/lol)), plott3(1:floor(count/lol),:));
title(['mRNA , dt = ', num2str(dt)]);
xlabel(Totaltid);
```

```
% legend('A','V','F','P','M','I','L','D','E','K','R','S','T','Y','H','C','N','Q','W','G',-1)
subplot(2,1,2);
plot(timeaxis(1:floor(count/lol)), plott4(1:floor(count/lol)));
title(['v , dt = ', num2str(dt)]);
xlabel(Totaltid);
figure(3);
set(3,'Position',[1 1 1152 796]);
subplot(2,1,1);
plot(timeaxis(1:floor(count/lol)), plott5(1:floor(count/lol),:));
title(['e , dt = ', num2str(dt)]);
xlabel(Totaltid);
% legend('A','V','F','P','M','I','L','D','E','K','R','S','T','Y','H','C','N','Q','W','G',-1)
subplot(2,1,2);
plot(timeaxis(1:floor(count/lol)), plott6(1:floor(count/lol),:));
title(['alpha , dt = ', num2str(dt)]);
xlabel(Totaltid);
% legend('A','V','F','P','M','I','L','D','E','K','R','S','T','Y','H','C','N','Q','W','G',-1)
figure(4);
set(4,'Position',[1 1 1152 796]);
subplot(2,1,1);
plot(timeaxis(1:floor(count/lol)), plott7(1:floor(count/lol),:));
title(['e(aaa+2) , dt = ', num2str(dt)]);
xlabel(Totaltid);
subplot(2,1,2);
plot(timeaxis(1:floor(count/lol)), plott8(1:floor(count/lol),:));
title(['ppGpp , dt = ', num2str(dt)]);
xlabel(Totaltid);
legend('ppGpp','spoT','relA');
figure(5);
set(5,'Position',[1 1 1152 796]);
subplot(2,1,1);
plot(timeaxis(1:floor(count/lol)), plott9(1:floor(count/lol),:));
title(['tRNA-komplex , dt = ', num2str(dt)]);
xlabel(Totaltid);
subplot(2,1,2);
plot(timeaxis(1:floor(count/lol)), plott10(1:floor(count/lol),:));
title(['mu , dt = ', num2str(dt)]);
xlabel(Totaltid);
%---------------
dX = zeros(length(plott2),aaa);
for i = 1:(length(plott2)-1)
    dX(i,:) = plott2(i+1,:) - plott2(i,:);
end
figure(42);
set(42,'Position',[1 1 1152 796]);
subplot(2,1,1);
plot(timeaxis(1:floor(count/lol)), dX(1:floor(count/lol),:));
title(['dX, dt = ', num2str(dt)]);
xlabel(Totaltid);
subplot(2,1,2);
plot(transtermer);
title(['transkriptionstermer']);
%---------------
figure(666)
set(666,'Position',[1 1 1152 796]);
subplot(3,3,1);
plot(timeaxis(1:floor(count/lol)), plott8(1:floor(count/lol),:));
title(['ppGpp ',num2str(ppGpp),' Hydro = ',num2str(hydrodel)]);
ylabel(['Växt = -',num2str(mu * ppGpp),' dppGpp = ',num2str(spoTdel + relAdel - hydrodel - mu * ppGpp)]);
xlabel(['RelA = ',num2str(relAdel),' spoT = ',num2str(spoTdel)]);
%  legend('ppGpp','spoT','relA','hydrodel',3);
subplot(3,3,2);
plot(timeaxis(1:floor(count/lol)), plott9(1:floor(count/lol),:));
title(['tRNA-komplex Minsta T = ',num2str(min(T))]);
xlabel(['Laddning = ',num2str(min(T3./Ttot)),' och ',num2str(max(T3./Ttot))]);
subplot(3,3,3);
plot(timeaxis(1:floor(count/lol)), plott2(1:floor(count/lol),:));
title(['Aminosyror']);
xlabel(['Minsta X = ',num2str(min(X))]);
subplot(3,3,4);
plot(timeaxis(1:floor(count/lol)), plott7(1:floor(count/lol)));
title(['e(aaa+2)']);
xlabel(['e(22) ',num2str(e(aaa+2))]);
subplot(3,3,5);
plot(timeaxis(1:floor(count/lol)), plott4(1:floor(count/lol)),timeaxis(1:floor(count/lol)), 1e4 .* plott10(1:floor(count/lol)),'r');
title(['v & mu*1e4']);
xlabel(['v = ',num2str(v),'mu = ',num2str(mu)]);
subplot(3,3,6);
plot(timeaxis(1:floor(count/lol)), plott5(1:floor(count/lol),:));
title(['e']);
subplot(3,3,7);
semilogy(timeaxis(1:floor(count/lol)), plott1(1:floor(count/lol),:));
title([' totalt: ',num2str(RNAP),' fritt: ',num2str(RNAPfri)]);
legend('RNAPtot','RNAPfritt',3);
xlabel(['totalt: ',num2str(RNAP),' fritt: ',num2str(RNAPfri)]);
subplot(3,3,8);
plot(timeaxis(1:floor(count/lol)), plott3(1:floor(count/lol),:));
title(['mRNA ',num2str(mRNA(aaa+1))]);
xlabel([num2str(current_time),' s ']);
subplot(3,3,9);
plot(timeaxis(1:floor(count/lol)), plott6(1:floor(count/lol),:));
title(['alpha']);
xlabel([num2str(procent),'procent klart med tidssteg ',num2str(dt)]);
%---------------

%----------totaltid---------
tiden = toc/60;
minuter = floor(tiden)
sekunder = floor(mod(tiden,1) * 60)
```

```
%--------------------------
save test_av_mer_decay;
```

The function which calculates the time derivative at every point:

```
function [ ppGpp, spoTdel, relAdel, hydrodel, de, dhydrospoT, mRNA, drRNA ] = Derivata( r, rRNA, tau, e, k, Kp, v, f, mu, ...
Kr, alphax, lambda, ppGpp, aaa, ...
    faktor, RelAterm, hydrospoT, Ttot, kr, Km, decay, boundrib )
Krp = 7.15e-7;                          %Dissociationskonst för bildande av RNAP-ppGpp.
%Jag testar att ändra den för att få mer ribbar i produktion e-7 -> e-6
A = [ 2.718281828     2.718281828     2.929152104     4.711491835     3.629330435 ];     %Ty P(mu) = A^(mu/B)
B = [  .4384931208e-3  .4384931208e-3  .4233530472e-3  .6796663525e-3  .3771514270e-3 ]; %Erhölls ur Bremers papper
Vmax = [ 110  110  33  1.5  33 ] ./ 60;
K_m   = [ 6600  1200  192  192  192 ];
RelA = 1e-6; %konst konc av RelA;
kcat = [ 100, 100, 5000 ]; %olika delar av ppGpp-systemet mer utspädningseffekter
khydro = 2e-4; %0.00555555555556; % per sec, k för att falla sönder till hydrolas = 1 per 3 min
RNAPfri=1e-6; %RNAPfri = 3e-6; Testa att ändra från 1e-7 för mer ribbar

P(1) = A(1)^(mu/B(1));
P(2) = A(2)^(mu/B(2));
P(3) = A(3)^(mu/B(3));
P(4) = A(4)^(mu/B(4));
P(5) = A(5)^(mu/B(5));

% e(aaa+2) ersättes av (e(aaa+2) - boundrib), ty e(aaa+2) är totala konc

for i = 1:aaa
    Regl(i) = alphax(i) * P(4) * Vmax(4) / ( 1 + K_m(4)/RNAPfri );  %mRNA-reglering med konstant RNAPfri
    mRNA(i) = Regl(i) / ( lambda + mu );                            %Substitut för mRNA
    de(i) = ( mRNA(i)*(r - boundrib) / (Kr + (r - boundrib)) ); %Decayet  - decay*e(i) ut, ty ej på init
end
Regl(aaa+1) = 0.03 * P(3) * Vmax(3) / ( 1 + K_m(3)/RNAPfri ); %spoT-mRNA-reglering med konstant RNAPfri
mRNA(aaa+1) = Regl(aaa+1) / ( lambda + mu );
de(aaa+1) = ( 1*mRNA(aaa+1)*(r - boundrib) / (Kr + (r - boundrib)) ); % Ändrat faktor 1 för mindre ppGppändringar
spoTterm = e(aaa+1) - hydrospoT;
dhydrospoT = - decay * hydrospoT;

RNAPppGpp = .5 * ( ppGpp + RNAPfri + Krp - sqrt( (ppGpp+RNAPfri+Krp).^2 - 4*ppGpp*RNAPfri) );  %Hur mkt komplex
del = RNAPppGpp / RNAPfri;   %Andel som är ppGpp-bundet
P1_prod = P(1) * Vmax(1) * ( 1 / ( 1 + K_m(1)/( RNAPfri*(1-del) ) ) + 1 / ( 1 + 1000*K_m(1)/( RNAPfri*del ) ) );
P2_prod = P(2) * Vmax(2) * ( 1 / ( 1 + K_m(2)/( RNAPfri*(1-del) ) ) + 1 / ( 1 + 1000*K_m(2)/( RNAPfri*del ) ) );
drRNA = P1_prod + P2_prod - mu * rRNA;
if rRNA < e(aaa+2)
    Regl(aaa+2) = 0;  %mRNA-reglering med konstant RNAPfri
else
    Regl(aaa+2) = 1 * P(3) * Vmax(3) / ( 1 + K_m(3)/RNAPfri ); %testing
end
mRNA(aaa+2) = Regl(aaa+2) / ( lambda + mu ); %reglering av ribprot
de(aaa+2) = ( mRNA(aaa+2)*(r - boundrib) / (Kr + (r - boundrib)) ); % Ribosomproteiner total

% dppGpp = spoTterm * kcat(2) / ( 1 + ppGpp/1e-4 ) + RelAterm * kcat(1) / ( 1 + ppGpp/1e-4 ) - hydrospoT * kcat(3) / ( 1 + 1e-6/ppGpp ) - mu * ppGpp;

%Stort uttryck som har infinitesimal Im-del, men ändå.
ppGpp = real(.3333333333e-6/mu*(-.4500000000e15*mu*spoTterm*kcat(2)*hydrospoT*kcat(3)-.4410000000e11*mu^2*spoTterm*kcat(2)-...
    .4500000000e15*mu*RelAterm*kcat(1)*hydrospoT*kcat(3)-.4410000000e11*mu^2*RelAterm*kcat(1)+.1470000000e15*mu*hydrospoT^2*kcat(3)^2+...
    .1529700000e11*mu^2*hydrospoT*kcat(3)-984851.*mu^3-.1000000000e19*hydrospoT^3*kcat(3)^3+150.*sqrt(-.2821800000e17*mu^2*hydrospoT^2*kcat(3)^2-...
    29403.*mu^4-.3237600000e17*mu^2*spoTterm^2*kcat(2)^2-.3000000000e25*hydrospoT^4*kcat(3)^4-.6475200000e17*mu^2*spoTterm*kcat(2)*RelAterm*kcat(1)+...
    .5705364000e17*mu^2*spoTterm*kcat(2)*hydrospoT*kcat(3)-.2364360000e22*mu*spoTterm*kcat(2)*hydrospoT^2*kcat(3)^2-...
    .6057612000e11*mu^3*spoTterm*kcat(2)-.3237600000e17*mu^2*RelAterm^2*kcat(1)^2-.6057612000e11*mu^3*RelAterm*kcat(1)-...
    .5821200000e11*mu^3*hydrospoT*kcat(3)+.5880000000e21*mu*hydrospoT^3*kcat(3)^3-.1200000000e22*mu*spoTterm^3*kcat(2)^3-...
    .1200000000e22*mu*RelAterm^3*kcat(1)^3-.3000000000e25*spoTterm^2*kcat(2)^2*hydrospoT^2*kcat(3)^2+...
    .5705364000e17*mu^2*RelAterm*kcat(1)*hydrospoT*kcat(3)-.2364360000e22*mu*RelAterm*kcat(1)*hydrospoT^2*kcat(3)^2+...
    .2940000000e22*mu*spoTterm^2*kcat(2)^2*hydrospoT*kcat(3)+.5880000000e22*mu*spoTterm*kcat(2)*RelAterm*kcat(1)*hydrospoT*kcat(3)-...
    .6000000000e25*spoTterm*kcat(2)*RelAterm*kcat(1)*hydrospoT^2*kcat(3)^2-.3600000000e22*mu*spoTterm*kcat(2)*RelAterm^2*kcat(1)^2+...
    .2940000000e22*mu*RelAterm^2*kcat(1)^2*hydrospoT*kcat(3)-.3600000000e22*mu*spoTterm^2*kcat(2)^2*RelAterm*kcat(1)+...
    .5880000000e25*spoTterm*kcat(2)*hydrospoT^3*kcat(3)^3-.3000000000e25*RelAterm^2*kcat(1)^2*hydrospoT^2*kcat(3)^2+...
    .5880000000e25*RelAterm*kcat(1)*hydrospoT^3*kcat(3)^3)*mu)^(1/3)+.3333333333e-6*(300000000.*mu*spoTterm*kcat(2)+...
    300000000.*mu*RelAterm*kcat(1)-98000000.*mu*hydrospoT*kcat(3)+9901.*mu^2+.1000000000e13*hydrospoT^2*kcat(3)^2)/mu/...
    (-.4500000000e15*mu*spoTterm*kcat(2)*hydrospoT*kcat(3)-.4410000000e11*mu^2*spoTterm*kcat(2)-.4500000000e15*mu*RelAterm*kcat(1)*hydrospoT*kcat(3)-...
    .4410000000e11*mu^2*RelAterm*kcat(1)+.1470000000e15*mu*hydrospoT^2*kcat(3)^2+.1529700000e11*mu^2*hydrospoT*kcat(3)-984851.*mu^3-...
    .1000000000e19*hydrospoT^3*kcat(3)^3+150.*sqrt(-.2821800000e17*mu^2*hydrospoT^2*kcat(3)^2-29403.*mu^4-.3237600000e17*mu^2*spoTterm^2*kcat(2)^2-...
    .3000000000e25*hydrospoT^4*kcat(3)^4-.6475200000e17*mu^2*spoTterm*kcat(2)*RelAterm*kcat(1)+.5705364000e17*mu^2*spoTterm*kcat(2)*hydrospoT*kcat(3)-...
    .2364360000e22*mu*spoTterm*kcat(2)*hydrospoT^2*kcat(3)^2-.6057612000e11*mu^3*spoTterm*kcat(2)-.3237600000e17*mu^2*RelAterm^2*kcat(1)^2-...
    .6057612000e11*mu^3*RelAterm*kcat(1)-.5821200000e11*mu^3*hydrospoT*kcat(3)+.5880000000e21*mu*hydrospoT^3*kcat(3)^3-...
    .1200000000e22*mu*spoTterm^3*kcat(2)^3-.1200000000e22*mu*RelAterm^3*kcat(1)^3-.3000000000e25*spoTterm^2*kcat(2)^2*hydrospoT^2*kcat(3)^2+...
    .5705364000e17*mu^2*RelAterm*kcat(1)*hydrospoT*kcat(3)-.2364360000e22*mu*RelAterm*kcat(1)*hydrospoT^2*kcat(3)^2+...
    .2940000000e22*mu*spoTterm^2*kcat(2)^2*hydrospoT*kcat(3)+.5880000000e22*mu*spoTterm*kcat(2)*RelAterm*kcat(1)*hydrospoT*kcat(3)-...
    .6000000000e25*spoTterm*kcat(2)*RelAterm*kcat(1)*hydrospoT^2*kcat(3)^2-.3600000000e22*mu*spoTterm*kcat(2)*RelAterm^2*kcat(1)^2+...
    .2940000000e22*mu*RelAterm^2*kcat(1)^2*hydrospoT*kcat(3)-.3600000000e22*mu*spoTterm^2*kcat(2)^2*RelAterm*kcat(1)+...
    .5880000000e25*spoTterm*kcat(2)*hydrospoT^3*kcat(3)^3-.3000000000e25*RelAterm^2*kcat(1)^2*hydrospoT^2*kcat(3)^2+...
    .5880000000e25*RelAterm*kcat(1)*hydrospoT^3*kcat(3)^3)*mu)^(1/3)-.3333333333e-6*(1000000.*hydrospoT*kcat(3)+101.*mu)/mu);
spoTdel = spoTterm * kcat(2);
relAdel = RelAterm * kcat(1);
hydrodel = - hydrospoT * kcat(3) / ( 1 + 1e-6/ppGpp );
```

Appendix B

MAPLE code for the calculations of the RNA promoter concentrations:

```
> alpha^(1/1800*1/beta) = 15.1;
```

$$\alpha^{(1/1800\,\frac{1}{\beta})} = 15.1$$

```
> alpha^(.1571348403e-2*1/beta) = 36;
```

$$\alpha^{(.001571348403\,\frac{1}{\beta})} = 54$$

```
> solve(alpha^(1/1800*1/.2046475232e-3*ln(alpha)) = 15.1 ,alpha);
```

$$.3678794412,\ 2.718281828$$

```
> solve(max(solve(alpha^(1/1800*1/.2046475232e-3*ln(alpha)) = 15.1
,alpha))^(.1571348403e-2*1/beta) = 36, beta);
```

$$.0004384931208$$

```
> solve(2.718281828^(.1571348403e-2*1/beta) = 36, beta);
```

$$.0004384931208$$

```
> alpha = 2.718281828;
```

$$\alpha = 2.718281828$$

```
> beta = .4384931208e-3;
```

$$\beta = .0004384931208$$

```
> A := max(solve(alpha^(1/1800*1/.2046475232e-3*ln(alpha)) = 9
,alpha));B = solve(A^(.1571348403e-2*1/beta) = 22, beta);
```

$$A := 2.458758388$$

$$B = .0004573453226$$

```
> evalf(root(1/180,3)/3);
```

.05903658717

```
> 1/10*150^(1/3)
```

Appendix C

The MAPLE code for calculating the concentration of ppGpp as it equilibriates fast:

```
> restart;

> ekv := 0 = spoTterm * kcat(2) / ( 1 + ppGpp/1e-4 ) + RelAterm
* kcat(1) / ( 1 + ppGpp/1e-4 ) - hydrospoT * kcat(3) / ( 1 + 1e-6/ppGpp
) - mu * ppGpp;;
```

$$ekv := 0 = \frac{spoTterm\,\mathrm{kcat}(2)}{1 + 10000.\,ppGpp} + \frac{RelAterm\,\mathrm{kcat}(1)}{1 + 10000.\,ppGpp} - \frac{hydrospoT\,\mathrm{kcat}(3)}{1 + \frac{.1\,10^{-5}}{ppGpp}} - \mu\,ppGpp$$

```
> ppGpp:=solve(ekv,ppGpp)[1];
```

$$\begin{aligned} ppGpp := .3333333333\,10^{-6}(&-.4500000000\,10^{15}\,\mu\,spoTterm\,\mathrm{kcat}(2)\,hydrospoT\,\mathrm{kcat}(3) \\ &- .4410000000\,10^{11}\,\mu^2\,spoTterm\,\mathrm{kcat}(2) \\ &- .4500000000\,10^{15}\,\mu\,RelAterm\,\mathrm{kcat}(1)\,hydrospoT\,\mathrm{kcat}(3) \\ &- .4410000000\,10^{11}\,\mu^2\,RelAterm\,\mathrm{kcat}(1) \\ &+ .1470000000\,10^{15}\,\mu\,hydrospoT^2\,\mathrm{kcat}(3)^2 \\ &+ .1529700000\,10^{11}\,\mu^2\,hydrospoT\,\mathrm{kcat}(3) - 984851.\,\mu^3 \\ &- .1000000000\,10^{19}\,hydrospoT^3\,\mathrm{kcat}(3)^3 + 150.\mathrm{sqrt}(\\ &- .3000000000\,10^{25}\,RelAterm^2\,\mathrm{kcat}(1)^2\,hydrospoT^2\,\mathrm{kcat}(3)^2 \\ &+ .5880000000\,10^{25}\,RelAterm\,\mathrm{kcat}(1)\,hydrospoT^3\,\mathrm{kcat}(3)^3 \\ &+ .5880000000\,10^{25}\,spoTterm\,\mathrm{kcat}(2)\,hydrospoT^3\,\mathrm{kcat}(3)^3 \\ &- .3000000000\,10^{25}\,spoTterm^2\,\mathrm{kcat}(2)^2\,hydrospoT^2\,\mathrm{kcat}(3)^2 \\ &- .3000000000\,10^{25}\,hydrospoT^4\,\mathrm{kcat}(3)^4 - 29403.\,\mu^4 \\ &- .3237600000\,10^{17}\,\mu^2\,spoTterm^2\,\mathrm{kcat}(2)^2 \\ &- .2364360000\,10^{22}\,\mu\,RelAterm\,\mathrm{kcat}(1)\,hydrospoT^2\,\mathrm{kcat}(3)^2 \\ &+ .5705364000\,10^{17}\,\mu^2\,RelAterm\,\mathrm{kcat}(1)\,hydrospoT\,\mathrm{kcat}(3) \\ &+ .2940000000\,10^{22}\,\mu\,RelAterm^2\,\mathrm{kcat}(1)^2\,hydrospoT\,\mathrm{kcat}(3) \\ &+ .5880000000\,10^{22}\,\mu\,spoTterm\,\mathrm{kcat}(2)\,RelAterm\,\mathrm{kcat}(1)\,hydrospoT\,\mathrm{kcat}(3) \\ &- .3600000000\,10^{22}\,\mu\,spoTterm^2\,\mathrm{kcat}(2)^2\,RelAterm\,\mathrm{kcat}(1) \\ &+ .2940000000\,10^{22}\,\mu\,spoTterm^2\,\mathrm{kcat}(2)^2\,hydrospoT\,\mathrm{kcat}(3) \\ &- .3600000000\,10^{22}\,\mu\,spoTterm\,\mathrm{kcat}(2)\,RelAterm^2\,\mathrm{kcat}(1)^2 \end{aligned}$$

()

$$
\begin{aligned}
&- .6475200000\,10^{17}\,\mu^2\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{RelAterm}\,\mathrm{kcat}(1)\\
&+ .5705364000\,10^{17}\,\mu^2\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{hydrospoT}\,\mathrm{kcat}(3)\\
&- .2364360000\,10^{22}\,\mu\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{hydrospoT}^2\,\mathrm{kcat}(3)^2\\
&- .6000000000\,10^{25}\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{RelAterm}\,\mathrm{kcat}(1)\,\mathit{hydrospoT}^2\,\mathrm{kcat}(3)^2\\
&- .6057612000\,10^{11}\,\mu^3\,\mathit{spoTterm}\,\mathrm{kcat}(2) - .6057612000\,10^{11}\,\mu^3\,\mathit{RelAterm}\,\mathrm{kcat}(1)\\
&- .2821800000\,10^{17}\,\mu^2\,\mathit{hydrospoT}^2\,\mathrm{kcat}(3)^2\\
&- .5821200000\,10^{11}\,\mu^3\,\mathit{hydrospoT}\,\mathrm{kcat}(3)\\
&+ .5880000000\,10^{21}\,\mu\,\mathit{hydrospoT}^3\,\mathrm{kcat}(3)^3\\
&- .1200000000\,10^{22}\,\mu\,\mathit{spoTterm}^3\,\mathrm{kcat}(2)^3\\
&- .1200000000\,10^{22}\,\mu\,\mathit{RelAterm}^3\,\mathrm{kcat}(1)^3\\
&- .3237600000\,10^{17}\,\mu^2\,\mathit{RelAterm}^2\,\mathrm{kcat}(1)^2)\mu)^{(1/3)}/\mu + .3333333333\,10^{-6}(\\
&.300000000\,10^{9}\,\mu\,\mathit{spoTterm}\,\mathrm{kcat}(2) + .300000000\,10^{9}\,\mu\,\mathit{RelAterm}\,\mathrm{kcat}(1)\\
&- .98000000\,10^{8}\,\mu\,\mathit{hydrospoT}\,\mathrm{kcat}(3) + 9901.\,\mu^2\\
&+ .1000000000\,10^{13}\,\mathit{hydrospoT}^2\,\mathrm{kcat}(3)^2)\Big/(\mu(\\
&- .4500000000\,10^{15}\,\mu\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{hydrospoT}\,\mathrm{kcat}(3)\\
&- .4410000000\,10^{11}\,\mu^2\,\mathit{spoTterm}\,\mathrm{kcat}(2)\\
&- .4500000000\,10^{15}\,\mu\,\mathit{RelAterm}\,\mathrm{kcat}(1)\,\mathit{hydrospoT}\,\mathrm{kcat}(3)\\
&- .4410000000\,10^{11}\,\mu^2\,\mathit{RelAterm}\,\mathrm{kcat}(1)\\
&+ .1470000000\,10^{15}\,\mu\,\mathit{hydrospoT}^2\,\mathrm{kcat}(3)^2\\
&+ .1529700000\,10^{11}\,\mu^2\,\mathit{hydrospoT}\,\mathrm{kcat}(3) - 984851.\,\mu^3\\
&- .1000000000\,10^{19}\,\mathit{hydrospoT}^3\,\mathrm{kcat}(3)^3 + 150.\mathrm{sqrt}(\\
&- .3000000000\,10^{25}\,\mathit{RelAterm}^2\,\mathrm{kcat}(1)^2\,\mathit{hydrospoT}^2\,\mathrm{kcat}(3)^2\\
&+ .5880000000\,10^{25}\,\mathit{RelAterm}\,\mathrm{kcat}(1)\,\mathit{hydrospoT}^3\,\mathrm{kcat}(3)^3\\
&+ .5880000000\,10^{25}\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{hydrospoT}^3\,\mathrm{kcat}(3)^3\\
&- .3000000000\,10^{25}\,\mathit{spoTterm}^2\,\mathrm{kcat}(2)^2\,\mathit{hydrospoT}^2\,\mathrm{kcat}(3)^2\\
&- .3000000000\,10^{25}\,\mathit{hydrospoT}^4\,\mathrm{kcat}(3)^4 - 29403.\,\mu^4\\
&- .3237600000\,10^{17}\,\mu^2\,\mathit{spoTterm}^2\,\mathrm{kcat}(2)^2\\
&- .2364360000\,10^{22}\,\mu\,\mathit{RelAterm}\,\mathrm{kcat}(1)\,\mathit{hydrospoT}^2\,\mathrm{kcat}(3)^2\\
&+ .5705364000\,10^{17}\,\mu^2\,\mathit{RelAterm}\,\mathrm{kcat}(1)\,\mathit{hydrospoT}\,\mathrm{kcat}(3)\\
&+ .2940000000\,10^{22}\,\mu\,\mathit{RelAterm}^2\,\mathrm{kcat}(1)^2\,\mathit{hydrospoT}\,\mathrm{kcat}(3)\\
&+ .5880000000\,10^{22}\,\mu\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{RelAterm}\,\mathrm{kcat}(1)\,\mathit{hydrospoT}\,\mathrm{kcat}(3)\\
&- .3600000000\,10^{22}\,\mu\,\mathit{spoTterm}^2\,\mathrm{kcat}(2)^2\,\mathit{RelAterm}\,\mathrm{kcat}(1)\\
&+ .2940000000\,10^{22}\,\mu\,\mathit{spoTterm}^2\,\mathrm{kcat}(2)^2\,\mathit{hydrospoT}\,\mathrm{kcat}(3)\\
&- .3600000000\,10^{22}\,\mu\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{RelAterm}^2\,\mathrm{kcat}(1)^2\\
&- .6475200000\,10^{17}\,\mu^2\,\mathit{spoTterm}\,\mathrm{kcat}(2)\,\mathit{RelAterm}\,\mathrm{kcat}(1)
\end{aligned}
$$

()

$$+ .5705364000\,10^{17}\,\mu^2\,spoTterm\,\mathrm{kcat}(2)\,hydrospoT\,\mathrm{kcat}(3)$$
$$- .2364360000\,10^{22}\,\mu\,spoTterm\,\mathrm{kcat}(2)\,hydrospoT^2\,\mathrm{kcat}(3)^2$$
$$- .6000000000\,10^{25}\,spoTterm\,\mathrm{kcat}(2)\,RelAterm\,\mathrm{kcat}(1)\,hydrospoT^2\,\mathrm{kcat}(3)^2$$
$$- .6057612000\,10^{11}\,\mu^3\,spoTterm\,\mathrm{kcat}(2) - .6057612000\,10^{11}\,\mu^3\,RelAterm\,\mathrm{kcat}(1)$$
$$- .2821800000\,10^{17}\,\mu^2\,hydrospoT^2\,\mathrm{kcat}(3)^2$$
$$- .5821200000\,10^{11}\,\mu^3\,hydrospoT\,\mathrm{kcat}(3)$$
$$+ .5880000000\,10^{21}\,\mu\,hydrospoT^3\,\mathrm{kcat}(3)^3$$
$$- .1200000000\,10^{22}\,\mu\,spoTterm^3\,\mathrm{kcat}(2)^3$$
$$- .1200000000\,10^{22}\,\mu\,RelAterm^3\,\mathrm{kcat}(1)^3$$
$$- .3237600000\,10^{17}\,\mu^2\,RelAterm^2\,\mathrm{kcat}(1)^2)\mu)^{(1/3)})$$
$$- \frac{.3333333333\,10^{-6}\,(.1000000\,10^{7}\,hydrospoT\,\mathrm{kcat}(3) + 101.\,\mu)}{\mu} \quad ()$$

```
> hydrospoT:=7e-7;
```

$$hydrospoT := .7\,10^{-6}$$

```
> RelAterm:=1.055e-7;
```

$$RelAterm := .1055\,10^{-6}$$

```
> spoTterm:=2e-7;
```

$$spoTterm := .2\,10^{-6}$$

```
> kcat(1):=100; kcat(2):=100; kcat(3):=k; k:=100;
```

$$\mathrm{kcat}(1) := 100$$

$$\mathrm{kcat}(2) := 100$$

$$\mathrm{kcat}(3) := k$$

$$k := 100$$

```
> mu:=5.6e-4;
```

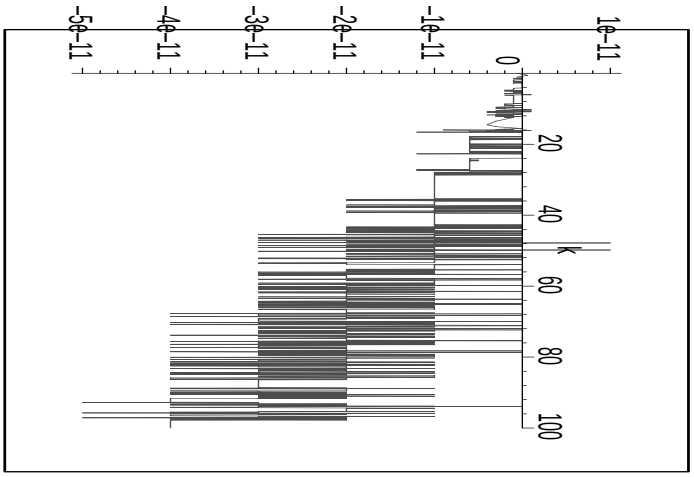
$$\mu := .00056$$

```
> kcat(3):= k; ppGpp;
```

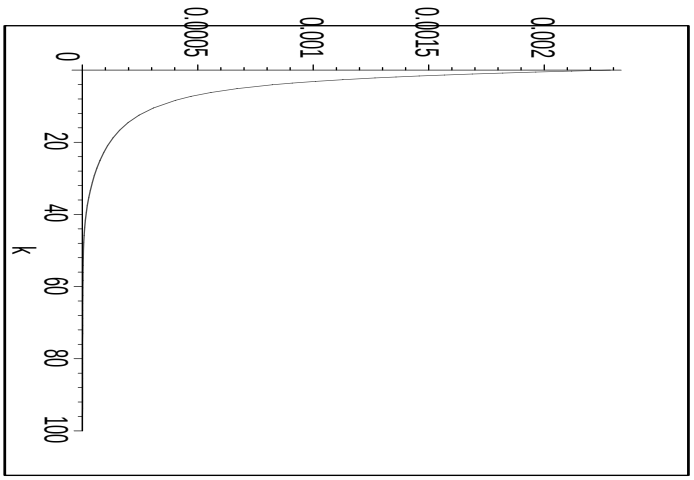
$$\mathrm{kcat}(3) := k$$

$$.0005952380953(-5.385662003\,k-.4226721236+.04033680000\,k^2-.3430000000\,k^3$$
$$+.08400\,\sqrt{1075.995079\,k-1391.779252\,k^2+61.72740504\,k^3-.7203000000\,k^4-19169.80325}$$
$$\sqrt{}^{(1/3)}+.0005952380953\,(5.135504954-.03841600000\,k+.4900000000\,k^2)/($$
$$-5.385662003\,k-.4226721236+.04033680000\,k^2-.3430000000\,k^3$$
$$+.08400\,\sqrt{1075.995079\,k-1391.779252\,k^2+61.72740504\,k^3-.7203000000\,k^4-19169.80325}$$
$$\sqrt{}^{(1/3)}-.0004166666666\,k-.00003366666666 \quad ()$$

```
> unassign('k'); plot(Im(ppGpp),k=0..100);
```

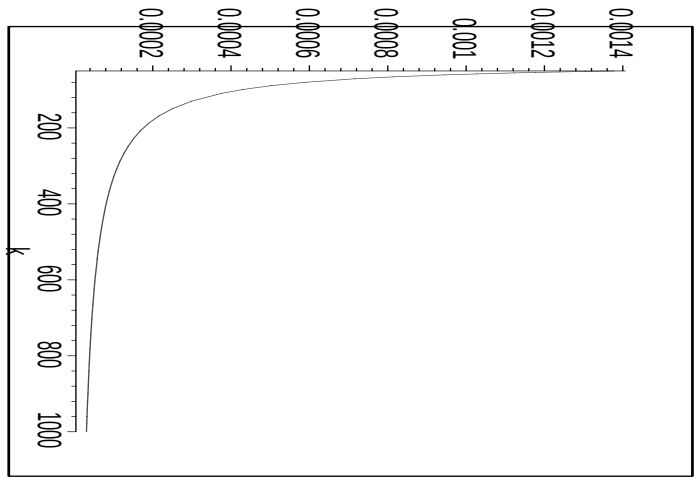



```
> unassign('k'); plot(Re(ppGpp),k=0..100);
```



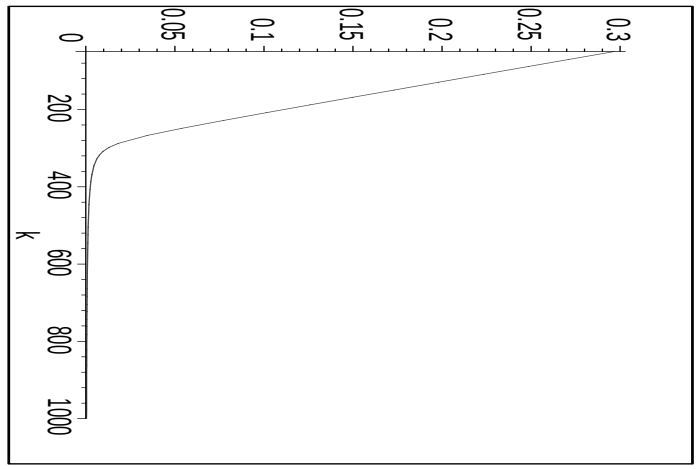
```
> kcat(1) := 10; plot(ppGpp,k=50..1000);
```

$$\mathrm{kcat}(1) := 10$$



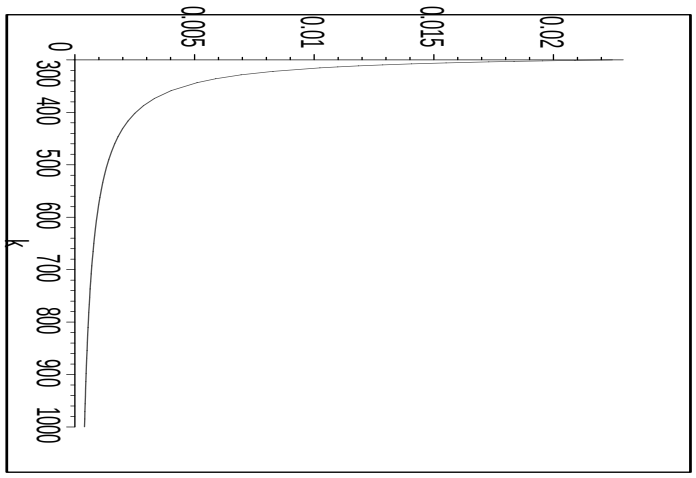

```
> kcat(2) := 1000; plot(ppGpp,k=50..1000);
```

$$\mathrm{kcat}(2) := 1000$$



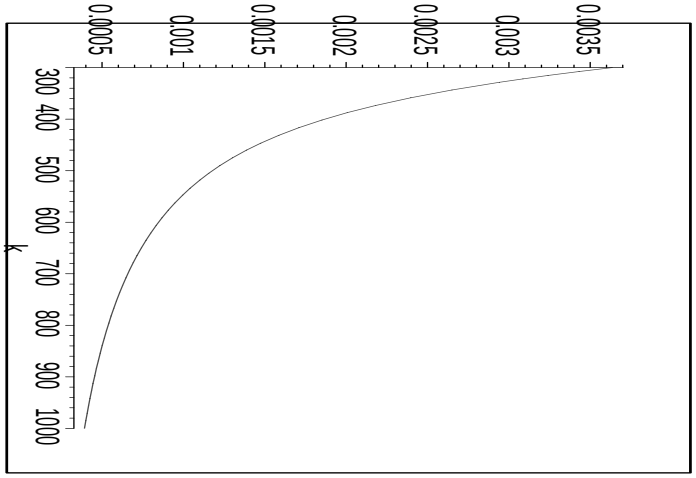

```
> mu:=1e-8 ; plot(ppGpp,k=300..1000);
```

$$\mu := .1\,10^{-7}$$




```
> mu:=1e-2 ; plot(ppGpp,k=300..1000);
```

$$\mu := .01$$



MAPLE code for the fast equilibrium of the ribosome-RelA-tRNA^c complex:

```
> restart;
> ekv1:=k[t]/(k[-b]+k[t])*AR*T+k[b]/(k[-b]+k[t])*AT*R=ART;
> ekv2:=k[a]/(k[-a]+k[t]*T)*A*R+k[-t]/(k[-a]+k[t]*T)*ART=AR;
> ekv3:=k[t]/(k[-t]+k[b]*R)*A*T+k[-b]/(k[-t]+k[b]*R)*ART=AT;
> ekv4:=k[-a]/(k[a]*R+k[t]*T)*AR+k[-t]/(k[a]*R+k[t]*T)*AT=A;
```

$$ekv1 := \frac{k_t\, AR\, T}{k_{-b} + k_t} + \frac{k_b\, AT\, R}{k_{-b} + k_t} = ART$$

$$ekv2 := \frac{k_a\, A\, R}{k_{-a} + k_t\, T} + \frac{k_{-t}\, ART}{k_{-a} + k_t\, T} = AR$$

$$ekv3 := \frac{k_t\, AT}{k_{-t} + k_b\, R} + \frac{k_{-b}\, ART}{k_{-t} + k_b\, R} = AT$$

$$ekv4 := \frac{k_{-a}\, AR}{k_a\, R + k_t\, T} + \frac{k_{-t}\, AT}{k_a\, R + k_t\, T} = A$$

```
> solve( {ekv1, ekv2, ekv3, ekv4}, {ART, AR, AT, A} );
```

$$\{AR = 0,\ AT = 0,\ A = 0,\ ART = 0\}$$

```
> K[t]:=k[-t]/k[t]=1e-5; K[a]:=k[-a]/k[a]=1e-5; K[b]:=k[-b]/k[b]=1e-5;
```

$$K_t := \frac{k_{-t}}{k_t} = .00001$$

$$K_a := \frac{k_{-a}}{k_a} = .00001$$

$$K_b := \frac{k_{-b}}{k_b} = .00001$$

```
> e1:=A*R*k[a]=k[-a]*AR; e2:=AR*T*k[t]=k[-t]*ART; e3:=A*T*k[t]=k[-t]*AT;
e4:=AT*R*k[b]=k[-b]*ART;
```

$$e1 := k_a\, A\, R = AR\, k_{-a}$$

$$e2 := k_t\, AR\, T = k_{-t}\, ART$$

$$e3 := k_t\, A\, T = AT\, k_{-t}$$

$$e4 := AT\, k_b\, R = k_{-b}\, ART$$

```
> solve( {e1, e2, e3, e4}, {ART, AR, AT, A} );
```

$$\{AR = 0,\ AT = 0,\ A = 0,\ ART = 0\}$$

```
> e5:= A+AR+AT+ART=A0; e6:= T+AT+ART=T0; e7:= R+AR+ART=R0;
```

$$e5 := A + AR + AT + ART = A0$$

$$e6 := T + AT + ART = T0$$

$$e7 := R + AR + ART = R0$$

```
> A:=solve(e5,A);
```

$$A := A0 - AR - AT - ART$$

```
> AT:=solve(e3,AT);
```

$$AT := \frac{k_t\,(A0 - AR - ART)\,T}{k_t\,T + k_{-t}}$$

```
> AR:=solve(e2,AR);
```

$$AR := \frac{k_{-t}\,ART}{k_t\,T}$$

```
> ART:=solve(e7,ART);
```

$$ART := \frac{(-R + R0)\,k_t\,T}{k_t\,T + k_{-t}}$$

```
> k[b]:=solve(e4,k[b]);
```

$$k_b := \frac{k_{-b}\,(-R + R0)}{(A0 + R - R0)\,R}$$

```
> k[t]:=solve(e6,k[t]);
```

$$k_t := \frac{k_{-t}\,(-T + T0)}{T\,(T - T0 + A0)}$$

```
> solve(K[t],T);
```

$$.5000000000\ T0 - .5000000000\ A0 - .5000000000\,10^{-5} + .5000000000\,10^{-5}\mathrm{sqrt}(.1000000000\,10^{11}\ T0^2 - .2000000000\,10^{11}\ T0\ A0 + 200000.\ T0 + .1000000000\,10^{11}\ A0^2 + 200000.\ A0 + 1.),\ .5000000000\ T0 - .5000000000\ A0 - .5000000000\,10^{-5} - .5000000000\,10^{-5}\mathrm{sqrt}(.1000000000\,10^{11}\ T0^2 - .2000000000\,10^{11}\ T0\ A0 + 200000.\ T0 + .1000000000\,10^{11}\ A0^2 + 200000.\ A0 + 1.) \quad ()$$

```
> T:=-.5000000000*A0-.5000000000e-5+.5000000000*T0+.5000000000e-5*sqrt(
.1000000000e11*A0^2+200000.*A0-.2000000000e11*A0*T0+1.+200000.*T0+.100
0000000e11*T0^2);
```

$$T := .5000000000\ T0 - .5000000000\ A0 - .5000000000\,10^{-5} + .5000000000\,10^{-5}\mathrm{sqrt}(.1000000000\,10^{11}\ T0^2 - .2000000000\,10^{11}\ T0\ A0 + 200000.\ T0 + .1000000000\,10^{11}\ A0^2 + 200000.\ A0 + 1.) \quad ()$$

```
> k[a]:=solve(e1, k[a]);
```

$$k_a := \frac{(-1.\ R + R0)\,k_{-a}}{(A0 + R - 1.\ R0)\,R}$$

```
> solve(K[a],R);
```

$$.5000000000\ R0 - .5000000000\,10^{-5} - .5000000000\ A0 + .5000000000\,10^{-5}\mathrm{sqrt}(.1000000000\,10^{11}\ R0^2 + 200000.\ R0 - .2000000000\,10^{11}\ R0\ A0 + 1. + 200000.\ A0 + .1000000000\,10^{11}\ A0^2),\ .5000000000\ R0 - .5000000000\,10^{-5} - .5000000000\ A0 - .5000000000\,10^{-5}\mathrm{sqrt}(.1000000000\,10^{11}\ R0^2 + 200000.\ R0 - .2000000000\,10^{11}\ R0\ A0 + 1. + 200000.\ A0 + .1000000000\,10^{11}\ A0^2) \quad ()$$

```
> R:=.5000000000*R0-.5000000000e-5-.5000000000*A0+.5000000000e-5*sqrt(.
1000000000e11*R0^2+200000.*R0-.2000000000e11*R0*A0+1.+200000.*A0+.1000
000000e11*A0^2);
```

$$R := .5000000000\ R0 - .5000000000\,10^{-5} - .5000000000\ A0 + .5000000000\,10^{-5}\mathrm{sqrt}(.1000000000\,10^{11}\ R0^2 + 200000.\ R0 - .2000000000\,10^{11}\ R0\ A0 + 1. + 200000.\ A0 + .1000000000\,10^{11}\ A0^2) \quad ()$$

```
> simplify(ART); simplify(AR);
```

$$.2500000000\,10^{-10}(100000.\,T0+100000.\,A0+1.-1.\mathrm{sqrt}(.1000000000\,10^{11}\,T0^2 - .2000000000\,10^{11}\,T0\,A0 + 200000.\,T0 + .1000000000\,10^{11}\,A0^2 + 200000.\,A0 + 1.))(100000.\,R0 + 1. + 100000.\,A0 - 1.\mathrm{sqrt}(.1000000000\,10^{11}\,R0^2 + 200000.\,R0 - .2000000000\,10^{11}\,R0\,A0 + 1. + 200000.\,A0 + .1000000000\,10^{11}\,A0^2))/A0 \quad ()$$

$$.2500000000\,10^{-10}(100000.\,R0+1.+100000.\,A0-1.\mathrm{sqrt}(.1000000000\,10^{11}\,R0^2 + 200000.\,R0 - .2000000000\,10^{11}\,R0\,A0 + 1. + 200000.\,A0 + .1000000000\,10^{11}\,A0^2))(-100000.\,T0 + 100000.\,A0 - 1. + \mathrm{sqrt}(.1000000000\,10^{11}\,T0^2 - .2000000000\,10^{11}\,T0\,A0 + 200000.\,T0 + .1000000000\,10^{11}\,A0^2 + 200000.\,A0 + 1.))/A0 \quad ()$$

```
> T0:=9.470137558362964e-006; A0:=4.507651491694952e-008; R0:=1e-6;
```

$$T0 := .94701375583629\,64\,10^{-5}$$

$$A0 := .45076514916949\,52\,10^{-7}$$

$$R0 := .1\,10^{-5}$$

```
> A; AT; ART; AR; A0;
```

$$.2107844662\,10^{-7}$$

$$.1991542131\,10^{-7}$$

$$.1983409695\,10^{-8}$$

$$.2099237305\,10^{-8}$$

$$.4507651491694952\,10^{-7}$$

```
> ART; AT; T; TO;
```

$$.1983409695\,10^{-8}$$

$$.1991542131\,10^{-7}$$

$$.9448238727\,10^{-5}$$

$$.9470137558362964\,10^{-5}$$

```
> ART; AR; R; RO;
```

$$.1983409695\,10^{-8}$$

$$.2099237305\,10^{-8}$$

$$.995917353\,10^{-6}$$

$$.1\,10^{-5}$$

```
> ART+AR;
```

$$.4082647000\,10^{-8}$$

Appendix D

The calculation of the flow of metabolites that runs through the system:

```
> restart;
```

```
> e1:= r*1/(Summan+(f(i)*1 / kr(i) * ( 1 + Km/(Ttot(i)-T(i)))))*f(i)
= X(i)*S0*k2 / (k2/k0 + X(i)*(Ks/T(i)));
```

$$e1 := \frac{r\,\mathrm{f}(i)}{Summan + \dfrac{\mathrm{f}(i)\,(1+\dfrac{Km}{\mathrm{Ttot}(i)-\mathrm{T}(i)})}{\mathrm{kr}(i)}} = \frac{\mathrm{X}(i)\,S0\,k2}{\dfrac{k2}{k0}+\dfrac{\mathrm{X}(i)\,Ks}{\mathrm{T}(i)}}$$

```
> e2:= r*1/(Summan+(f(i)*1 / kr(i) * ( 1 + Km/(Ttot(i)-T(i)))))*f(i)
= ( k(i)*e(i) / ( 1 + X(i)/Kp ) + f(i)*faktor(i) );
```

$$e2 := \frac{r\,\mathrm{f}(i)}{Summan + \dfrac{\mathrm{f}(i)\,(1+\dfrac{Km}{\mathrm{Ttot}(i)-\mathrm{T}(i)})}{\mathrm{kr}(i)}} = \frac{\mathrm{k}(i)\,\mathrm{e}(i)}{1+\dfrac{\mathrm{X}(i)}{Kp}} + \mathrm{f}(i)\,\mathrm{faktor}(i)$$

```
> e3:=( k(i)*e(i) / ( 1 + X(i)/Kp ) + f(i)*faktor(i) ) = X(i)*S0*k2
/ (k2/k0 + X(i)*(Ks/T(i)));
```

$$e3 := \frac{\mathrm{k}(i)\,\mathrm{e}(i)}{1+\dfrac{\mathrm{X}(i)}{Kp}} + \mathrm{f}(i)\,\mathrm{faktor}(i) = \frac{\mathrm{X}(i)\,S0\,k2}{\dfrac{k2}{k0}+\dfrac{\mathrm{X}(i)\,Ks}{\mathrm{T}(i)}}$$

```
> X(i):=solve(e1,X(i));
```

$$\begin{aligned}\mathrm{X}(i) := {} & -r\,\mathrm{kr}(i)\,\mathrm{f}(i)\,k2\,\mathrm{T}(i)\,(\mathrm{Ttot}(i)-\mathrm{T}(i))/(k0(r\,\mathrm{kr}(i)\,\mathrm{f}(i)\,\mathrm{Ttot}(i)\,Ks - r\,\mathrm{kr}(i)\,\mathrm{f}(i)\,\mathrm{T}(i)\,Ks \\ & - S0\,k2\,\mathrm{T}(i)\,Summan\,\mathrm{kr}(i)\,\mathrm{Ttot}(i) + S0\,k2\,\mathrm{T}(i)^2\,Summan\,\mathrm{kr}(i) \\ & - S0\,k2\,\mathrm{T}(i)\,\mathrm{f}(i)\,\mathrm{Ttot}(i) + S0\,k2\,\mathrm{T}(i)^2\,\mathrm{f}(i) - S0\,k2\,\mathrm{T}(i)\,\mathrm{f}(i)\,Km)) \quad \text{(D-1)}\end{aligned}$$

```
> e2;
```

$$\begin{aligned}\frac{r\,\mathrm{f}(i)}{Summan + \dfrac{\mathrm{f}(i)\,(1+\dfrac{Km}{\mathrm{Ttot}(i)-\mathrm{T}(i)})}{\mathrm{kr}(i)}} = {} & \mathrm{k}(i)\,\mathrm{e}(i)/(1-r\,\mathrm{kr}(i)\,\mathrm{f}(i)\,k2\,\mathrm{T}(i)\,(\mathrm{Ttot}(i)-\mathrm{T}(i))/(k0(\\ & r\,\mathrm{kr}(i)\,\mathrm{f}(i)\,\mathrm{Ttot}(i)\,Ks - r\,\mathrm{kr}(i)\,\mathrm{f}(i)\,\mathrm{T}(i)\,Ks - S0\,k2\,\mathrm{T}(i)\,Summan\,\mathrm{kr}(i)\,\mathrm{Ttot}(i) \\ & + S0\,k2\,\mathrm{T}(i)^2\,Summan\,\mathrm{kr}(i) - S0\,k2\,\mathrm{T}(i)\,\mathrm{f}(i)\,\mathrm{Ttot}(i) + S0\,k2\,\mathrm{T}(i)^2\,\mathrm{f}(i) \\ & - S0\,k2\,\mathrm{T}(i)\,\mathrm{f}(i)\,Km)Kp)) + \mathrm{f}(i)\,\mathrm{faktor}(i) \quad \text{(D-2)}\end{aligned}$$

```
> r/(Summan+f(i)/kr(i)*(1+Km/(Ttot(i)-X(i)*Ks*k0*(k(i)*e(i)*Kp+f(i)*fak
tor(i)*Kp+f(i)*faktor(i)*X(i))/k2/(-k(i)*e(i)*Kp-f(i)*faktor(i)*Kp-f(i
)*faktor(i)*X(i)+X(i)*S0*k0*Kp+X(i)^2*S0*k0))))*f(i) =
> (i)*e(i)/(1+X(i)/Kp)+f(i)*faktor(i)
```

$$\frac{r\,\mathrm{f}(i)/(Summan+\mathrm{f}(i)(1+Km/(\mathrm{Ttot}(i)-\mathrm{X}(i)\,Ks\,k0\,(\mathrm{k}(i)\,\mathrm{e}(i)\,Kp+\mathrm{f}(i)\,\mathrm{faktor}(i)\,Kp+\mathrm{f}(i)\,\mathrm{faktor}(i)\,\mathrm{X}(i))}{k2\,(-\mathrm{k}(i)\,\mathrm{e}(i)\,Kp-\mathrm{f}(i)\,\mathrm{faktor}(i)\,Kp-\mathrm{f}(i)\,\mathrm{faktor}(i)\,\mathrm{X}(i)+\mathrm{X}(i)\,S0\,k0\,Kp+\mathrm{X}(i)^2\,S0\,k0)}\\ ))/\mathrm{kr}(i))=\frac{\mathrm{k}(i)\,\mathrm{e}(i)}{1+\frac{\mathrm{X}(i)}{Kp}}+\mathrm{f}(i)\,\mathrm{faktor}(i) \quad \text{(D-3)}$$

```
> restart;
> r*v*f(i) = (k(i)*e(i)+f(i)*faktor(i))/(1+X(i)/Kp);
```

$$r\,v\,\mathrm{f}(i)=\frac{\mathrm{k}(i)\,\mathrm{e}(i)+\mathrm{f}(i)\,\mathrm{faktor}(i)}{1+\frac{\mathrm{X}(i)}{Kp}}$$

```
> r:=faktor(i)*v; v:=.15;
```

$$r := .15\,\mathrm{faktor}(i)$$

$$v := .15$$

```
> solve(r*v*f(i) = (k(i)*e(i)+f(i)*faktor(i))/(1+X(i)/Kp),v);
```

$$\frac{(\mathrm{k}(i)\,\mathrm{e}(i)+\mathrm{f}(i)\,\mathrm{faktor}(i))\,Kp}{r\,\mathrm{f}(i)\,(Kp+\mathrm{X}(i))}$$

```
> ( k*e + f*faktor + decay * f * ( slask ) ) / e(aaa+2) /f;
```

$$\frac{k\,e+f\,faktor+decay\,f\,slask}{\mathrm{e}(aaa+2)\,f}$$