MSc Mathematics

Track: Biomedical Mathematics

Master thesis

Computer-assisted validation of solutions to differential-algebraic systems in systems biology

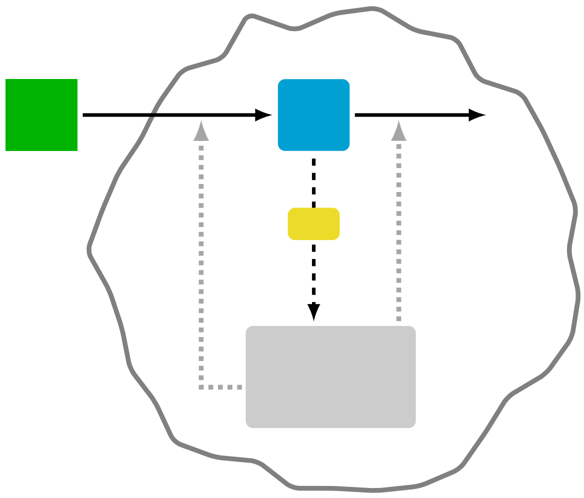
by

Thomas Antoon Rooijakkers

March 28, 2018

Supervisor: prof.dr. Jan Bouwe van den Berg

Second examiner: dr. Robert Planqué



Department of Mathematics
Faculty of Science

VRIJE UNIVERSITEIT AMSTERDAM

Abstract

Today, computer simulations are a vital tool in the mathematical analysis of dynamical systems. Simulations allow people to imagine the, often complex, dynamics of (nonlinear) systems. Besides being a great tool for inspiration, simulations can also be used to test and reject hypotheses. Making those numerical computations rigorous truly unleashes the power of numerics.

In this thesis, we consider the validation of numerically computed solutions to differential-algebraic systems in the domain of systems biology. We treat a particular toy model of the qORAC framework, in the form of a semi-explicit differential-algebraic system, to illustrate a computer-assisted validation technique that proves the existence of an exact, and unique solution to the dynamical system, within an explicit range of the numerically computed solution.

Title: Computer-assisted validation of solutions to differential-algebraic systems in systems biology
Author: Thomas Antoon Rooijakkers, thomas.rooijakkers@vu.nl, 2175967
Supervisor: prof.dr. Jan Bouwe van den Berg
Second examiner: dr. Robert Planqué
Date: March 28, 2018

Department of Mathematics
VU Amsterdam
De Boelelaan 1081, 1081 HV Amsterdam
`http://www.math.vu.nl/`

Contents

1. Introduction

Natural environments are inhabited by a vast and diverse selection of microbial species. There is a constant battle between these microbes for both space and resources. To survive and thrive in this microbial jungle with fluctuating nutrient concentrations, microbes need to grow fast, and preferably faster than their competitors, to outcompete the competition.

Enzymes are a sub-class of proteins with the ability to accelerate chemical reactions. Nearly all microbial reactions are catalysed by enzymes [6, 8]. The reaction rates of these so called *enzymatic reactions* increases or decreases by changing the enzyme concentrations. A protein allocation perspective can be used to explain, and understand, many aspects of microbial growth as indicated in recent work [4, 10, 14, 15].

Enzymatic resources are finite in supply, forcing the cells to be economical; cells need to synthesise the right amount of enzymes, at the right time. This enzymatic tuning is an everlasting process as there is a constant fluctuation in the nutrient levels of the cell's environment. Obtaining and maintaining high growth rates thus requires the microbial cells to continuously adapt their enzyme levels on the changing environmental conditions.

How cells tackle this tuning challenge to sustain high reaction rates is still unclear. However, there have been attempts to model this tuning process [7, 11, 12]. In this thesis we consider one of these attempts discussed in [11]. Dr. R. Planqué, et. al. present a framework called qORAC, showing that there is a general solution to this aforementioned tuning problem. In particular they discuss how cells can, with the help of simple regulation, maintain an optimal metabolism.

***q*ORAC** qORAC stands for Specific Flux (q) Optimisation by Robust Adaptive Control. This adaptive control theory assumes that cells can only sense changes in internal metabolites. These internal changes can be used as a sensor for the external changes.

Consider the regulation of galactose metabolism in yeast, visualised in Figure 1.1. In this example a cell needs to distribute a finite amount of enzymatic resources over the four pathway enzymes. One of these pathway enzymes catalyses the galactose import reaction. The amount of enzymatic resources to invest in this import reaction depends on the external galactose concentration. We mentioned above that it is assumed that a cell is only capable to sense internal concentration changes. However, an increase in $[\mathrm{Gal}_{\mathrm{out}}]$ results in an increase in $[\mathrm{Gal}_{\mathrm{in}}]$ when the enzyme concentration that catalyses the import reaction is fixed. In this case $[\mathrm{Gal}_{\mathrm{in}}]$ can act as a sensor concentration and allows the cell to alter the enzyme allocation when external conditions change.

It has been shown that, in yeast, the inner galactose metabolite acts as a sensor [9]. More specifically, the sensor metabolite ($\mathrm{Gal}_{\mathrm{in}}$) binds to a transcription factor (Gal3p) which in turn influences the gene expression and thereby the enzyme concentrations in the cell.

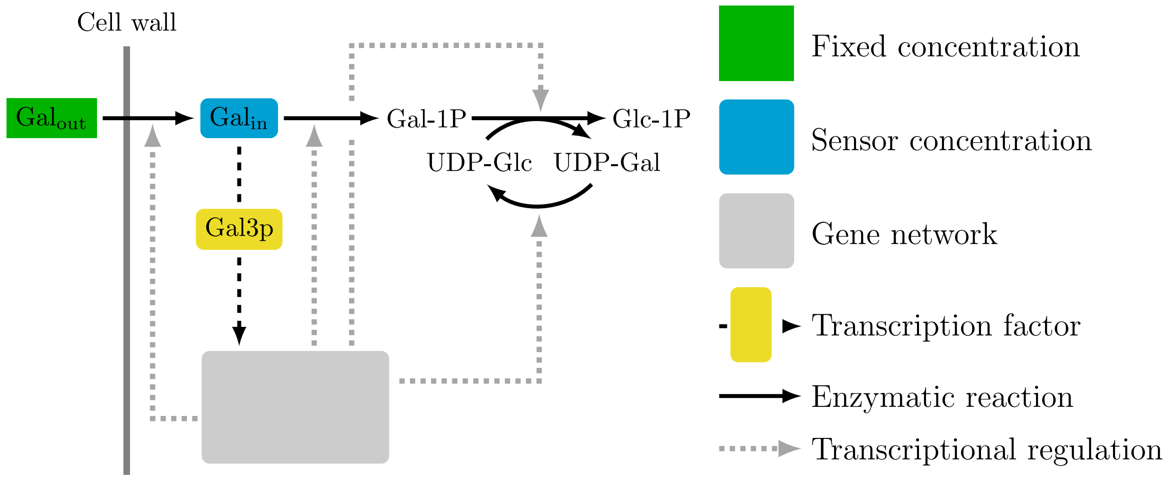


Figure 1.1.: The regulation of galactose metabolism in yeast. Abbreviations: Gal$_{\text{out}}$, outer galactose; Gal$_{\text{in}}$, inner galactose; Gal-1P, galactose-1-phosphate; Glc-1P, glucose-1-phosphate; UDP-Glc, urine-diphosphate-glucose; UDP-Gal, urine-diphosphate-galactose; Gal3p, Gal3 protein. This visualisation has been adapted from [11].

Following the qORAC framework, we obtain combination of differential and algebraic equations that together model the enzyme tuning problem. We refer to the obtained system as a differential-algebraic system.

Differential-algebraic systems Consider a system of equations of the form

$$G(t, x(t), \dot{x}(t)) = 0, \tag{1.1}$$

where $x : [a, b] \to \mathbb{R}^n$ is a vector of n dependent variables $x(t) = (x_1(t), \dots, x_n(t))$ and G a vector containing n equations, $G = (G_1, \dots, G_n) : \mathbb{R}^{2n+1} \to \mathbb{R}^n$. An equation of the form (1.1) is called a differential-algebraic equation/system (DAE) if the Jacobian $\frac{\partial G}{\partial \dot{x}}$ of G is singular. If the Jacobian of G is nonsingular then we can solve system (1.1) for $\dot{x}$ to obtain a system of ordinary differential equations (ODEs). DAEs differ from ODEs in that DAEs contain equations that are algebraic; i.e. a DAE is not completely solvable using only the derivatives of all components of x, since they might not all appear. In this thesis we consider a special case of differential-algebraic systems. We consider the so-called semi-explicit DAEs, which can be written as

$$\dot{x} = f(t, x, y), \tag{1.2a}$$
$$0 = g(t, x, y), \tag{1.2b}$$

where x denotes the differential variables and y denotes the algebraic variables. Clearly, Equation (1.2a) represents the differential part of the system, whereas Equation (1.2b) represents the algebraic part of the system.

Rigorous computing We can analyse a differential-algebraic system by simulating the dynamics of the system with the help of a computer. These computer-assisted simulations can result in very detailed information, creating new insights, motivations and inspirations for further research. Furthermore, ensuring that the computed results are in fact genuinely a solution to the posed problem can greatly improve the mathematical insights as this allows such solutions to be used in theorems.

INTLAB We use INTLAB to be able to develop numerical methods that yield reliable results and make the performed computations rigorous. INTLAB, for INTerval LABoratory, is a MATLAB/Octave toolbox for reliable computing that has been developed by prof.dr. S.M. Rump at Hamburg University of Technology, Germany[13]. INTLAB extends MATLAB by introducing interval arithmetic. This allows us to put bounds on rounding and measurement errors in mathematical computations and, thereby, enables us to develop numerical methods that yield reliable results. INTLAB uses intervals, or ranges, instead of values to perform numerical computations.

Goal In [5], a computer-assisted validation technique for ODEs is introduced. The goal of this thesis is to extend the theory of [5] to DAEs. To illustrate this extension, we consider a simplified toy model of the qORAC framework and solve this toy model numerically. With the assistance of a computer, we want to prove that an exact, and unique, solution to this toy model lies within balls (with certain radii) around the numerically determined solution.

We consider the following toy model

$$\underline{v} \xrightarrow{V_1} u \xrightarrow{V_2} \varnothing, \tag{1.3}$$

consisting of only two reactions and two metabolites $\underline{v}$, and u. Note that $\underline{v}$ represents the outer fixed concentration, where the underline is used to denote that it is fixed. Furthermore, u represents the inner sensor metabolite. The system has been visualised in Figure 1.2.

The reaction rates of system (1.3), $V_1,\ V_2$, are given by

$$V_1 = e_1 f_1(\underline{v}, u), \quad V_2 = e_2 f_2(u),$$

where V_1 and V_2 depend on distinct enzymes e_1 and e_2, respectively. Furthermore, the functions f_1 and f_2 are of the form

$$f_1(\underline{v}, u) = \frac{\underline{v} - K_{\mathrm{eq}} u}{K_1 + u + \underline{v}}, \quad f_2(u) = \frac{u}{L + u},$$

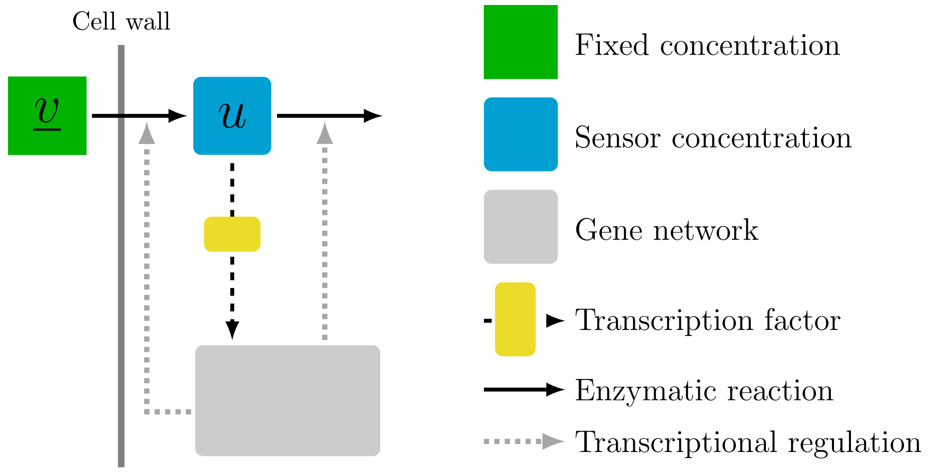


Figure 1.2.: Visualisation of toy model (1.3), in line with Figure 1.1.

which can be recognised as Michealis-Menten kinetics [1] with parameters K_1, K_{eq}, L, $\underline{v} \in \mathbb{R}^+$. Following the theory of qORAC [11], we obtain the system

$$\begin{cases} \dot{u} = e_1 f_1(\underline{v}, u) - e_2 f_2(u), \\ \dot{e}_1 = E_1(\phi, u) - e_1, \\ \dot{e}_2 = E_2(\phi, u) - e_2, \end{cases} \tag{1.4}$$

where

$$E_1(\phi, u) = \frac{1}{f_1(\phi, u)} \frac{1}{O(\phi, u)}, \quad E_2(\phi, u) = 1 - E_1(\phi, u).$$

We complete the system by adding an equation for ϕ. To do so, we introduce the *objective function*

$$O(\phi, u) = \frac{1}{f_1(\phi, u)} + \frac{1}{f_2(u)}.$$

and add the algebraic equation $\frac{\partial O}{\partial u}(\phi, u) = 0$ to system (1.4) to obtain the complete differential-algebraic system

$$\begin{cases} \dot{u} = e_1 f_1(\underline{v}, u) - e_2 f_2(u), \\ \dot{e}_1 = E_1(\phi, u) - e_1, \\ \dot{e}_2 = E_2(\phi, u) - e_2, \\ 0 = \dfrac{\partial O}{\partial u}(\phi, u). \end{cases} \tag{1.5}$$

We want to solve the initial value problem of system (1.5) and verify the solution.

Approach To achieve the goal of finding a ball around the numerically computed solution containing an exact, and unique solution, and thereby validating the numerical solution, we proceed as follows, we

1. reformulate the dynamic problem into a root-finding problem, $F(x) = 0$, in some infinite dimensional space,
2. define a Newton-like map T such that fixed points of T correspond to roots of F,
3. find conditions that guarantee the existence of a fixed point of T,
4. verify the conditions using a combination of numerical and analytical arguments,
5. obtain balls with radii r around the numerically computed solution containing an exact, and unique solution.

For the first step of the approach, we decide to work in a space of Taylor coefficients. We consider the Taylor series expansions

$$u(t) = \sum_{n=0}^{\infty} a_n t^n, \quad e_1(t) = \sum_{n=0}^{\infty} b_n t^n, \text{ etc.}$$

to transform the dynamic problem of interest into a root-finding problem.

In Chapter 2, we motivate the second step, and discuss Newton's Method. We construct a Newton-like map T and find conditions guaranteeing the existence of a fixed point of T. In other words, we discuss all the necessary theoretical material to be able to perform steps 2 and 3, of the above mentioned approach, in Chapter 2. In Chapter 3, we work out two basic examples. An ODE example, with only one variable, and a DAE example with two variables (one differential and one algebraic variable). These basic examples enable us to introduce notation that proves useful when considering system (1.5) in Chapter 4. In this chapter, we consider the analytical arguments of step 4. Having treated all theoretical details, and all necessary preparations, we are ready to continue with the computational results of Chapter 5. We finish off this thesis with a conclusional chapter, Chapter 6, where we sum up the main results of this thesis and make suggestions for future research.

2. Theory

In this chapter, we guide the reader through the theoretical part of computer-assisted proofs in differential-algebraic systems. Our goal is to prove that an exact, and unique, solution to a differential-algebraic system lies within a ball with an explicit radius around a numerically computed solution of the system.

To achieve this goal, we first transform the problem of finding a solution to a differential-algebraic system into a root-finding problem in an infinite dimensional setting. After this transformation we define a Newton-like map. In Section 2.3, we introduce conditions under which this Newton-like map is a uniform contraction, and provide conditions under which the existence of a fixed point of the Newton-like map is guaranteed. We introduce alternative, more practical conditions, based on the conditions of Section 2.3, in Section 2.4. We finish this chapter by introducing the weighted ℓ^1-norm in Section 2.5, and defining a product space with conditions guaranteeing the existence of a fixed point in this product space in Section 2.6.

2.1. Newton's Method

A root-finding algorithm is an algorithm that finds (approximate) roots, or zeroes, of continuous functions. Newton's method is a numerical iteration method used to determine approximate roots of a differentiable function, and is thus an example of a root-finding algorithm.

To demonstrate the Newton's method, consider a continuous, differentiable, real-valued, nonlinear function $g \in C^1(\mathbb{R}, \mathbb{R})$. If we want to find a root of this function, i.e. $g(x) = 0$, we can implement the Newton's method hoping to find a good approximate root. To start the method we need the function g, its derivative g' and an initial guess, x_0, for the root. Next, we iterate the so-called Newton scheme,

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)},$$

until we reach a sufficiently accurate value, e.g. one can decide to stop iterating when the difference between two consecutive iterations is bounded by some preset threshold, i.e. $|x_n - x_{n-1}| < \delta$, where δ represents a small threshold value. We can visualise this iteration process, as has been done in Figure 2.1, where we observe that the tangent of the function g evaluated at $(x_0, g(x_0))$ intersects the x-axis at $(x_1, 0)$. For the next iteration we follow the same procedure starting in x_1, etc.

We can extend this univariate Newton's method to a multivariate Newton's method. We follow the motivation of [5] and let $f \in C^1(\mathbb{R}^N, \mathbb{R}^N)$ be a nonlinear function. Assume

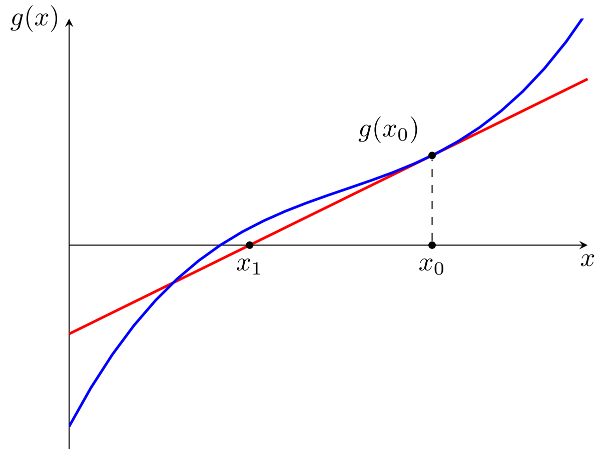


Figure 2.1.: Newton iteration visualised. The blue line visualises the function $g(x)$, the red line visualises the tangent of g in x_0.

that we want to find a solution to the nonlinear problem $f(x) = 0$, numerically. We define the multivariate Newton scheme $\tilde{T}(x_n)$ as

$$x_{n+1} = \tilde{T}(x_n) := x_n - (Df(x_n))^{-1} f(x_n), \tag{2.1}$$

where D denotes the Jacobian. We again start with some initial guess x_0 and hope that, by iterating the scheme, we obtain a sequence $\{x_n\}$ that converges to a root of f rapidly.

Why does this work? Consider the Jacobian of the multivariate Newton scheme $\tilde{T}(x_n)$ of Equation (2.1)

$$D\tilde{T}(x)y = y - (D[(Df(x))^{-1}]y)f(x) - (Df(x))^{-1}Df(x)y = (D[(Df(x))^{-1}]y)f(x).$$

We observe that $D\tilde{T}(\hat{x})y = 0$, for any $\hat{x}$ such that $f(\hat{x}) = 0$, provided $Df(\hat{x})$ is invertible. Thus, if $Df(\hat{x})$ is invertible, then the norm of the Jacobian of the multivariate Newton scheme $\|D\tilde{T}(x)\|$ is small for any x near the fixed point $\hat{x}$ of $\tilde{T}$, i.e. $\tilde{T}$ is a contraction mapping with strong contraction rate.

Remark that there is still room to alter the definition of the multivariate Newton scheme to obtain a less strong contraction mapping whilst maintaining a contraction. For instance, we can consider the following scheme

$$x_{n+1} = x_n - (Df(x_0))^{-1} f(x_n),$$

so that we do not need to recompute the inverse of the Jacobian, $(Df(x_n))^{-1}$, at every iteration.

2.2. Newton-like map

As mentioned in the introduction of this chapter, we first transform our dynamic problem of interest into a root-finding problem, $F(x) = 0$, in an infinite dimensional setting. Let us first introduce some notation; let X and X' be Banach spaces, let $F : X \to X'$ be a Fréchet differentiable map, assume that $\bar{x} \in X$ is an approximate, numerically determined root of F, and assume that there exists a left-inverse $(DF(\bar{x}))^{-1}$ of $DF(\bar{x})$. Using the aforementioned definitions and assumptions, we can define a Newton-like map

$$\hat{T}(x) = x - (DF(\bar{x}))^{-1} F(x).$$

Observe that fixed points of the Newton-like map $\hat{T}$ correspond to roots of F, since we assumed invertibility of $DF(\bar{x})$. However, we must note that inverting the Jacobian $DF(\bar{x})$ is often too hard in an infinite dimensional setting. To counteract this issue, we approximate the Jacobian $DF(\bar{x})$ by a linear operator $A^\dagger$

$$A^\dagger : X \to X',$$

and define A as an approximate left-inverse of $A^\dagger$.

To start our approach to approximate $DF(\bar{x})$, we first need to introduce some notation. We want to construct a finite, N-dimensional, truncated problem for some $N \in \mathbb{N}$. By π_N and π'_N we denote the projection of X and X' onto finite, N-dimensional subspaces X_N and X'_N, respectively. Furthermore, we denote the natural embedding of X_N into X by ι. Lastly, let $F_N : X_N \to X'_N$ denote the finite-dimensional mapping from X_N to X'_N. Using the introduced notation we obtain a finite-dimensional truncated problem;

$$0 = F_N(x_N) := \pi'_N F(\iota x_N). \tag{2.2}$$

Next, we numerically search for a solution $\bar{x}_N$ to the finite-dimensional problem (2.2). Having found an approximate root of $F_N(x_N)$, and thereby an approximate root $\bar{x} := \iota \bar{x}_N$ of $F(x)$, we compute the Jacobian of $F_N(x_N)$ and numerically invert it; i.e.

$$\begin{aligned} A_N^\dagger &: X_N \to X'_N, & A_N &: X'_N \to X_N, \\ A_N^\dagger &:= DF_N(\bar{x}_N), & A_N &\approx (DF_N(\bar{x}_N))^{-1}. \end{aligned} \tag{2.3}$$

Lastly, we need to extend the operator A_N to an injective operator A. Remark that making such an extension is problem specific. However, this extension needs to be properly balanced in simplicity and accuracy. We want an extension of A to be simple enough to perform some explicit analysis, whilst being accurate enough to be an approximate inverse.

Now that we defined the approximate inverse A, we define the Newton-like operator $T(x)$;

$$T(x) := x - AF(x). \tag{2.4}$$

Remark that if A is an injective operator, which we explicitly assumed, then fixed points of T are in one-to-one correspondence to the roots of F.

2.3. Theorem

In this section, we introduce conditions that guarantee that the Newton-like map T of Section 2.2 is a contraction map on a *closed* ball around the approximate fixed point

$\bar{x}$. We denote this ball of radius $\hat{r}$, and center $\bar{x}$ by $B_{\hat{r}}(\bar{x})$. Furthermore, we show that under the same conditions the existence of a unique, fixed point of T within this ball is guaranteed. The conditions are summarised in Theorem 2.1, obtained from [5].

Theorem 2.1 (T has a unique fixed point)**.** *Let T be a Fréchet differentiable map from a Banach space X to itself, such that for all $r > 0$*

$$\|T(\bar{x}) - \bar{x}\|_X \leq Y, \tag{2.5}$$

$$\|DT(x)\|_{B(X)} \leq Z(r) \quad \text{for all} \;\; x \in B_r(\bar{x}), \tag{2.6}$$

for some $Y \in \mathbb{R}^+$ and some function $Z : \mathbb{R}^+ \to \mathbb{R}^+$.
If there exists an $\hat{r} > 0$ such that

$$Y + \hat{r}Z(\hat{r}) < \hat{r}, \tag{2.7}$$

then T has a unique fixed point in $B_{\hat{r}}(\bar{x})$.

2.3.1. Proof of Theorem 2.1

An approach to prove Theorem 2.1 can be subdivided in three parts, namely

1. Verify that $B_{\hat{r}}(\bar{x})$ gets mapped into itself,
2. Show that T is a contraction on $B_{\hat{r}}(\bar{x})$,
3. Apply the Banach contraction mapping principle.

We visualised parts 1 and 2 in Figures 2.2 and 2.3, respectively. For the first part, we want to show that the norm of the difference between mapped value $T(x)$ and the center of the ball $\bar{x}$ is less than the radius, $\hat{r}$, of the ball, for arbitrary x:

$$\|T(x) - \bar{x}\|_X < \hat{r} \quad \text{for all} \;\; x \in B_{\hat{r}}(\bar{x}).$$

For the second part, we want to show that for two arbitrary elements x_1, $x_2 \in B_{\hat{r}}(\bar{x})$ the norm of the difference in their mapped value is less than the norm of the initial difference. In other words, there exists a $0 \leq k < 1$ such that for all x_1, $x_2 \in B_{\hat{r}}(\bar{x})$

$$\|T(x_1) - T(x_2)\|_X \leq k\|x_1 - x_2\|_X.$$

The combination of these two parts proves that T is a contraction mapping on $B_{\hat{r}}(\bar{x})$. For proving parts 1 and 2, we use the Mean Value Theorem for Banach spaces (Lemma 2.2), stated below.

Lemma 2.2 (Mean Value Theorem for Banach spaces)**.** *Let U be an open subset of X. Suppose that $T : U \to Y$ is Fréchet differentiable at any point of U. Let a, $b \in U$ such that*

$$[a, b] := \{ta + (1 - t)b : t \in [0, 1]\} \subset U,$$

then

$$\|T(b) - T(a)\|_Y \leq \sup_{c \in [a,b]} \|DT(c)\|\|b - a\|_U.$$

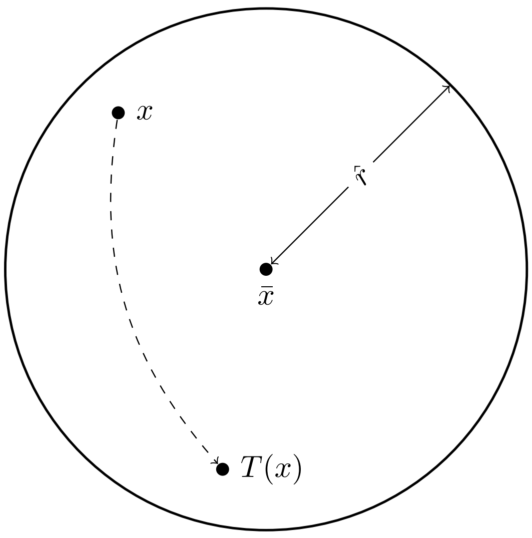


Figure 2.2.: We visualise part 1, where the circle represents the ball $B_{\tilde{r}}(\bar{x})$. We want to show that an arbitrary $x \in B_{\tilde{r}}(\bar{x})$ gets mapped into $B_{\tilde{r}}(\bar{x})$, i.e. $T(x) \in B_{\tilde{r}}(\bar{x})$, for all $x \in B_{\hat{r}}(\bar{x})$.

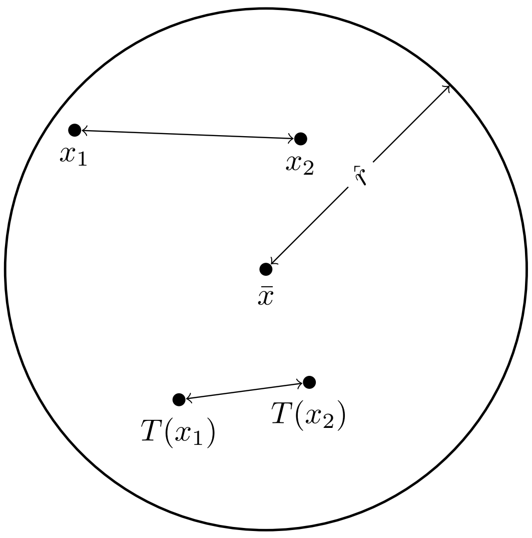


Figure 2.3.: We visualise part 2, where the circle represents the ball $B_{\tilde{r}}(\bar{x})$. We want to show that arbitrary x_1, $x_2 \in B_{\tilde{r}}(\bar{x})$ contract under the mapping T, i.e. $\|T(x_1) - T(x_2)\|_X \leq k\|x_1 - x_2\|_X$, with $0 \leq k < 1$, for all x_1, $x_2 \in B_{\hat{r}}(\bar{x})$.

After proving that T is a contraction mapping on $B_{\hat{r}}(\bar{x})$, we apply the Banach contraction mapping principle (Lemma 2.3) on T to finish the proof of Theorem 2.1. We have thereby proven that T has a unique, fixed point under conditions (2.5) to (2.7).

Lemma 2.3 (Banach contraction mapping principle)**.** *Let U be a (non-empty) closed subset of a Banach space X. If $T : U \to U$ is a contraction, then T has a unique fixed point.*

Remark 1. We do not prove Lemmas 2.2 and 2.3 here. For proofs of these lemmas we refer to textbooks on (nonlinear) analysis and fixed point theory, e.g. [2, 3].

Proof Theorem 2.1.
Part 1 ($B_{\hat{r}}(\bar{x})$ gets mapped into itself)
Let $x \in B_{\hat{r}}(\bar{x})$, then, using the triangle inequality, we obtain

$$\|T(x) - \bar{x}\|_X \leq \|T(x) - T(\bar{x})\|_X + \|T(\bar{x}) - \bar{x}\|_X.$$

Observe that, by Equation (2.5), $\|T(\bar{x}) - \bar{x}\|_X \leq Y$. By Equation (2.6) and the mean value theorem for Banach spaces (Lemma 2.2), we find

$$\|T(x) - T(\bar{x})\|_X \leq \sup_{x' \in B_{\hat{r}}(\bar{x})} \|DT(x')\|_{B(X)} \|x - \bar{x}\|_X \leq Z(\hat{r})\hat{r},$$

where we used that $\|x-\bar{x}\|_X \leq \hat{r}$, since $x \in B_{\hat{r}}(\bar{x})$. By Equation (2.7), we conclude that

$$\|T(x)-\bar{x}\|_X \leq Z(\hat{r})\hat{r} + Y < \hat{r}.$$

Part 2 (T is a contraction on $B_{\hat{r}}(\bar{x})$)

Let x_1, $x_2 \in B_{\hat{r}}(\bar{x})$ be arbitrary, then by Equation (2.6) and the mean value theorem for Banach spaces (Lemma 2.2)

$$\|T(x_1)-T(x_2)\|_X \leq \sup_{x' \in B_{\hat{r}}(\bar{x})} \|DT(x')\|_{B(X)} \|x_1 - x_2\|_X \leq Z(\hat{r})\|x_1-x_2\|_X.$$

Now observe that, by Equation (2.7), $Y + Z(\hat{r})\hat{r} < \hat{r}$ with $Y \in \mathbb{R}^+$. From this inequality we obtain that the contraction rate $Z(\hat{r})$ is less than one and conclude that

$$\|T(x_1)-T(x_2)\|_X \leq k\|x_1-x_2\|_X,$$

with $k = Z(\hat{r}) < 1$.

Part 3 (Apply Banach contraction mapping principle)

By part 1, we obtained that $T : B_{\hat{r}}(\bar{x}) \to B_{\hat{r}}(\bar{x})$, where $B_{\hat{r}}(\bar{x}) \subseteq X$ denotes the closed ball of radius $\hat{r}$ centered at $\bar{x}$. By part 2, we obtained that T is a contraction. The Banach contraction mapping principle (Lemma 2.3) completes the proof. □

2.4. Alternative conditions

In this section, we follow [5] to reformulate conditions (2.5) to (2.7) of Theorem 2.1 into conditions that are simpler to work with. In practice the function $Z(r)$ of Theorem 2.1 is an r-dependent polynomial. Assuming that $Z(r)$ is a polynomial in r we define the *radii polynomial*.

Definition 2.4. The *radii polynomial* is defined as

$$p(r) := Y + [Z(r)-1]r. \tag{2.8}$$

The radii polynomial enables us to reformulate condition (2.7) of Theorem 2.1 into a clearer goal. To prove that there is a unique fixed point of T in $B_{\hat{r}}(\bar{x})$ we need to find a positive $\hat{r}$ such that the radii polynomial is negative, i.e. $p(\hat{r}) < 0$ with $\hat{r} > 0$.

In Section 2.2, we set up a Newton-like map T (Equation (2.4)). Using this Newton-like map we can reformulate bounds (2.5) and (2.6) of Theorem 2.1 in terms of F. Let us first substitute $T(x) = x - AF(x)$ in bounds (2.5) and (2.6) to obtain

$$\|AF(\bar{x})\|_X \leq Y, \tag{2.9}$$
$$\|D[x - AF(x)]\|_{B(X)} \leq Z(r) \quad \text{for all } x \in B_r(\bar{x}). \tag{2.10}$$

Next, we observe that Equation (2.10) can be split into three different components with the help of the linear operator $A^\dagger \approx DF(\bar{x})$. This split proves useful in Chapter 3 where we discuss various examples. By adding and subtracting terms and subsequently applying the triangle inequality, we find

$$\begin{aligned}\|D[x - AF(x)]\|_{B(X)} &= \|I - DAF(x)\|_{B(X)}\\ &\leq \|I - AA^\dagger\|_{B(X)} + \|A[A^\dagger - DF(\bar{x})]\|_{B(X)}\\ &\quad + \|A[DF(\bar{x}) - DF(x)]\|_{B(X)}.\end{aligned}$$

Having split Equation (2.10), we have reformulated conditions (2.5) and (2.6) as

$$\|AF(\bar{x})\|_X \leq Y, \tag{2.9}$$
$$\|I - AA^\dagger\|_{B(X)} \leq Z^0, \tag{2.10a}$$
$$\|A[A^\dagger - DF(\bar{x})]\|_{B(X)} \leq Z^1, \tag{2.10b}$$
$$\|A[DF(\bar{x}) - DF(x)]\|_{B(X)} \leq Z^2(r) \quad \text{for all } x \in B_r(\bar{x}), \tag{2.10c}$$

where Y, Z^0, $Z^1 \in \mathbb{R}^+$, $Z^2 : \mathbb{R}^+ \to \mathbb{R}^+$, and where we set $Z(r) = Z^0 + Z^1 + Z^2(r)$. Observe that only Z^2 depends on r since Z^2 bounds the only x-dependent expression. Substituting $Z(r)$ into Equation (2.8) we obtain the radii polynomial

$$p(r) = Y + [Z^0 + Z^1 + Z^2(r) - 1]r. \tag{2.11}$$

Remark 2. Note that, the smallest $\hat{r}$ such that $p(\hat{r}) < 0$ gives the best bound on where the fixed point of T is, whilst the largest $\hat{r}$ satisfying $p(\hat{r}) < 0$ gives the strongest uniqueness result.

2.5. The weighted ℓ^1-norm

In Chapter 3, we decide to work in a space of Taylor coefficients. The differential-algebraic equations that we are working with are analytic, implying that the solution of the differential-algebraic systems will be analytic and that the Taylor coefficients decay geometrically. In this section, we introduce the weighted ℓ^1-norm, $\|\cdot\|_{\ell^1_\nu}$, and prove some useful properties of this norm to be used in the corresponding space ℓ^1_ν.

Definition 2.5. The weighted ℓ^1-norm is defined as

$$\|a\|_{\ell^1_\nu} := \sum_{k=0}^{\infty} |a_k|\nu^k,$$

where $\nu > 0$ denotes the weight.

The weight ν mentioned in the Definition 2.5 has to be chosen in the computer-assisted part of the proving process. In Chapter 3, we observe that by setting ν smaller we can decrease the size of (parts of) the bounds on Y, Z^0, Z^1, $Z^2(r)$ enabling us to increase the chances of finding an $\hat{r}$ such that $p(\hat{r}) < 0$.

We first show that the space ℓ^1_ν is a Banach space in Theorem 2.6

Theorem 2.6. *The space ℓ^1_ν, corresponding to the weighted ℓ^1-norm of Definition 2.5, is a Banach space.*

Proof. We need to show that ℓ^1_ν is complete, i.e. every Cauchy sequence in ℓ^1_ν converges to an element in ℓ^1_ν. Let $(X_n) \subset \ell^1_\nu$ be a Cauchy sequence with $X_n = (x_i^{(n)})_{i=0}^\infty$, $X_n \in \ell^1_\nu$, i.e. for given $\epsilon > 0$ there exists an $N \in \mathbb{N}$ such that for every $n,\ m \geq N$

$$\|X_n - X_m\|_{\ell^1_\nu} < \epsilon.$$

Next, we observe that for every $i \geq 0$,

$$\left|x_i^{(n)} - x_i^{(m)}\right| \nu^i \leq \sum_{k=0}^\infty \left|x_k^{(n)} - x_k^{(m)}\right| \nu^k = \|X_n - X_m\|_{\ell^1_\nu},$$

with $\nu > 0$. Let $\epsilon_i := \nu^i \epsilon$ then

$$\left|x_i^{(n)} - x_i^{(m)}\right| \nu^i \leq \epsilon_i, \text{ implies } \left|x_i^{(n)} - x_i^{(m)}\right| \leq \epsilon.$$

Hence, for every $i \geq 0$, the sequence $(x_i^{(n)})$ is Cauchy in $\mathbb{R}$. The space $\mathbb{R}$ with the standard metric is a complete metric space [16], and thus the sequence $(x_i^{(n)})$ converges to some $x_i \in \mathbb{R}$. We set $X = (x_k)_{k=0}^\infty$ and suspect that X is the limit in ℓ^1_ν of the Cauchy sequence (X_n).

We first need to show that $X \in \ell^1_\nu$. Observe the following:

$$\begin{aligned}\|X\|_{\ell^1_\nu} = \sum_{k=0}^\infty |x_k| \nu^k = \lim_{K\to\infty} \sum_{k=0}^K |x_k| \nu^k &= \lim_{K\to\infty} \left(\sum_{k=0}^K \left| \lim_{n\to\infty} x_k^{(n)} \right| \nu^k \right) \\ &= \lim_{K\to\infty} \left(\lim_{n\to\infty} \sum_{k=0}^K \left| x_k^{(n)} \right| \nu^k \right),\end{aligned}$$

where we note that we can interchange the limit with the sum of a finite number of real numbers without difficulties. Furthermore, we observe that there exists an M such that

$$\sum_{k=0}^K \left|x_k^{(n)}\right| \nu^k \leq \sum_{k=0}^\infty \left|x_k^{(n)}\right| \nu^k = \|X_n\|_{\ell^1_\nu} < M,$$

for all n, since X_n is a Cauchy sequence. Taking the limit $n \to \infty$ we find

$$\sum_{k=0}^K |x_k| \nu^k = \lim_{n\to\infty} \sum_{k=0}^K \left|x_k^{(n)}\right| \nu^k < M.$$

Now, since K is arbitrary we conclude that

$$\|X\|_{\ell^1_\nu} = \sum_{k=0}^\infty |x_k| \nu^k \leq M,$$

and hence $X \in \ell^1_\nu$.

Lastly, we show that $\|X_n - X\|_{\ell^1_\nu} \to 0$ as $n \to \infty$. Let $\epsilon > 0$ be given, then there is an N such that $\|X_n - X_m\|_{\ell^1_\nu} < \epsilon$ for all n, $m \geq N$. Hence, for every $K \geq 0$ we have

$$\sum_{k=0}^{K} \left| x_k^{(n)} - x_k^{(m)} \right| \nu^k \leq \|X_n - X_m\|_{\ell^1_\nu} < \epsilon, \qquad \text{for all } n, m \geq N.$$

Next, we fix $n > N$ and K, and send $m \to \infty$ to find

$$\sum_{k=0}^{K} \left| x_k^{(n)} - x_k \right| \nu^k \leq \epsilon.$$

However, since this holds for every K, we obtain

$$\|X_n - X\|_{\ell^1_\nu} = \sum_{k=0}^{\infty} \left| x_k^{(n)} - x_k \right| \leq \epsilon,$$

for $n > N$.

We summarise $\|X_n - X\|_{\ell^1_\nu} \to 0$ with $X \in \ell^1_\nu$, hence the space ℓ^1_ν is complete (and thereby a Banach space). □

To increase readability, we introduce the discrete convolution product, $(a * b)$, in Definition 2.7 and prove that the ℓ^1_ν-norm of this convolution product is bounded by the product of the norms in Proposition 2.8; i.e. the convolution product satisfies the Banach algebra property

$$a, b \in \ell^1_\nu : \|a * b\|_{\ell^1_\nu} \leq \|a\|_{\ell^1_\nu} \|b\|_{\ell^1_\nu}.$$

Definition 2.7. Let a, $b \in \ell^1_\nu$. The components of the *convolution product* $a * b \in \ell^1_\nu$ are defined by

$$(a * b)_k = \sum_{k'=0}^{k} a_{k'} b_{k-k'},$$

where $k \geq 0$.

Proposition 2.8. *The convolution product, as defined in Definition 2.7, satisfies the Banach algebra property; i.e.*

$$a, b \in \ell^1_\nu : \|a * b\|_{\ell^1_\nu} \leq \|a\|_{\ell^1_\nu} \|b\|_{\ell^1_\nu}.$$

Proof. Let a, $b \in \ell^1_\nu$ and consider the norm of the convolution product between a and b

$$\begin{aligned}
\|a * b\|_{\ell^1_\nu} &= \sum_{k=0}^{\infty} \left| \sum_{k'=0}^{k} a_{k'} b_{k-k'} \right| \nu^k \\
&\leq \sum_{k=0}^{\infty} \sum_{k'=0}^{k} |a_{k'} b_{k-k'}| \nu^k \\
&= \sum_{k=0}^{\infty} \sum_{k'=0}^{k} |a_{k'}| \nu^{k'} |b_{k-k'}| \nu^{k-k'} \\
&= \sum_{k'=0}^{\infty} |a_{k'}| \nu^{k'} \sum_{k-k'=0}^{\infty} |b_{k-k'}| \nu^{k-k'} \\
&= \|a\|_{\ell^1_\nu} \|b\|_{\ell^1_\nu},
\end{aligned}$$

where we used the triangle inequality. □

Next, we want to obtain an expression for the Jacobian of the convolution product, which will prove useful in practice. In Proposition 2.9 we obtain this expression.

Proposition 2.9. *Let a, b, $v \in \ell^1_\nu$. The components of the Jacobian of the convolution product, as defined in Definition 2.7, are given by*

$$D_a[(a * b)_k]v = (b * v)_k, \quad D_b[(a * b)_k]v = (a * v)_k,$$

where $k \geq 0$ and D_a and D_b denote the derivatives with respect to a and b, respectively.

Proof. Without loss of generality, we consider only $D_a[(a * b)_k]v$. Observe that

$$\frac{\partial (a * b)_k}{\partial a_{k'}} = \begin{cases} b_{k-k'} & 0 \leq k' \leq k, \\ 0 & \text{otherwise.} \end{cases}$$

In vector notation, this can be represented as

$$D_a[(a * b)_k]v = \begin{bmatrix} b_k & b_{k-1} & \dots & b_0 & 0 & \dots \end{bmatrix} v = (b * v)_k,$$

where $\begin{bmatrix} b_k & b_{k-1} & \dots & b_0 & 0 & \dots \end{bmatrix}$ denote an infinite dimensional row vector, and v denotes an infinite dimensional column vector. □

Lastly, we observe that we can obtain an upper bound on the operator norm $\|\cdot\|_{B(\ell^1_\nu, \ell^1_\nu)}$ in Proposition 2.10.

Proposition 2.10. *Let $G \in B(\ell^1_\nu, \ell^1_\nu)$, the operator norm $\|G\|_{B(\ell^1_\nu, \ell^1_\nu)}$ is bounded by*

$$\|G\|_{B(\ell^1_\nu, \ell^1_\nu)} \leq \sup_{m \geq 0} \frac{1}{\nu^m} \sum_{k=0}^{\infty} |G_{km}| \nu^k,$$

where $G_{km} := G(\delta^m)_k$ and δ denotes the Kronecker delta,

$$\delta^m_k = \begin{cases} 1 & k = m, \\ 0 & k \neq m. \end{cases} \tag{2.12}$$

Proof. Let $G \in B(\ell^1_\nu, \ell^1_\nu)$, and observe that by linearity we can write

$$\|G(a)\|_{\ell^1_\nu} = \sum_{k=0}^{\infty} |G(a)_k|\nu^k = \sum_{k=0}^{\infty}\sum_{m=0}^{\infty} |G(\delta^m)_k a_m|\nu^k,$$

using the Kronecker delta (2.12) and $a \in \ell^1_\nu$. We can bound this as follows:

$$\sum_{m=0}^{\infty}\sum_{k=0}^{\infty} |G(\delta^m)_k a_m|\nu^k \le \sum_{m=0}^{\infty} |a_m|\nu^m \sup_{m\ge 0} \frac{1}{\nu^m} \sum_{k=0}^{\infty} |G(\delta^m)_k|\nu^k = \|a\|_{\ell^1_\nu} \sup_{m\ge 0} \frac{1}{\nu^m} \sum_{k=0}^{\infty} |G_{km}|\nu^k.$$

We complete the proof by using the definition of an operator norm

$$\|G\|_{B(\ell^1_\nu,\ell^1_\nu)} = \sup\{\|G(a)\|_{\ell^1_\nu} : a \in \ell^1_\nu \text{ with } \|a\|_{\ell^1_\nu} = 1\} \le \sup_{m\ge 0} \frac{1}{\nu^m} \sum_{k=0}^{\infty} |G_{km}|\nu^k.$$

□

2.6. Product space

In this section we extend Theorem 2.1 to a specific product space to be used in Chapter 3 and 4. We establish conditions that, once satisfied, guarantee the existence of a unique fixed point of T. Let us first introduce some notation;

1. by $\mathcal{X}$ we denote the ordered list of the n variables/unknowns in ℓ^1_ν,
2. by ρ_i, $\sigma_i \in \mathcal{X}$ we denote the i-th element of $\mathcal{X}$ when the order is relevant,
3. by $X = \prod_{\rho\in\mathcal{X}}^n X_\rho = (\ell^1_\nu)^n$ we denote the product space of Banach spaces $X_\rho = \ell^1_\nu$,
4. by π^ρ we denote the projection of the variable ρ onto its corresponding space X_ρ,
5. by $\mathcal{B}_r(\bar{x}) := \{x \in X : \|\pi^\sigma(x - \bar{x})\|_{\ell^1_\nu} \le r_\sigma \text{ for all } \sigma \in \mathcal{X}\}$ we denote the hyperrectangle, where $r = (r_\sigma)_{\sigma\in\mathcal{X}} \in (\mathbb{R}^+)^n$ denotes the radii and $\bar{x} = (\bar{x}_\sigma)_{\sigma\in\mathcal{X}} \in X$.

Furthermore we define the n-dimensional radii polynomial in Definition 2.11 based on Definition 2.4.

Definition 2.11. The n-dimensional *radii polynomial* is defined as

$$p(r) := Y + (Z(r) - I)r,$$

with identity matrix I, vector $Y = (Y_{\sigma_1}, \dots, Y_{\sigma_n})$, and matrix Z, where $Z_{ij} = Z_{\sigma_i\rho_j}$, and variable $r = (r_{\sigma_1}, \dots, r_{\sigma_n}) \in (\mathbb{R}^+)^n$. The n-dimensional radii polynomial is a column vector of n polynomials.

Instead of using integers as indices, we introduced the ordered list $\mathcal{X}$. This makes it easier to trace which bounds and polynomials belong to which variables. In Theorem 2.12, we summarise the conditions that guarantee the existence of a unique fixed point of T.

Theorem 2.12. *Let $X = (\ell^1_\nu)^n$ denote the product space of Banach spaces ℓ^1_ν and $\mathcal{X}$ the ordered list of n variables. Let T be a Fréchet differentiable map from X to itself, such that for every $r = (r_\sigma)_{\sigma\in\mathcal{X}} \in (\mathbb{R}^+)^n$,*

$$\|\pi^\sigma[T(\bar{x}) - \bar{x}]\|_{\ell^1_\nu} \le Y_\sigma \qquad \text{for all } \sigma \in \mathcal{X}, \tag{2.13}$$

$$\|D_\rho \pi^\sigma T(x)\|_{B(\ell^1_\nu,\ell^1_\nu)} \le Z_{\sigma\rho}(r) \qquad \text{for all } x \in \mathcal{B}_r(\bar{x}),\ \sigma,\rho \in \mathcal{X}, \tag{2.14}$$

for some scalars $Y_\sigma \in \mathbb{R}^+$, and functions $Z_{\sigma\rho} : \mathbb{R}^+ \to \mathbb{R}^+$, with σ, $\rho \in \mathcal{X}$. If there exists an $\hat{r} \in (\mathbb{R}^+)^n$ such that the n-dimensional radii polynomial of Definition 2.11 is negative component-wise for some $\hat{r} \in (\mathbb{R}^+)^n$, i.e. $p(\hat{r}) < 0$ component-wise, then T has a unique fixed point in $\mathcal{B}_{\hat{r}}(\bar{x})$.

To prove this theorem, we follow a similar approach as for the proof of Theorem 2.1. We first show that T maps $\mathcal{B}_r(\bar{x})$ into itself. Next, we show that T is a contraction mapping, and eventually we apply the Banach contraction mapping principle (Lemma 2.3).

Proof.
Part 1 ($\mathcal{B}_r(\bar{x})$ gets mapped into itself)
We introduce notation for the projected balls

$$B_{\hat{r}_\sigma}(\bar{x}_\sigma) := \{y \in \ell^1_\nu : \|y - \bar{x}_\sigma\|_{\ell^1_\nu} \le \hat{r}_\sigma\},$$

then $\mathcal{B}_{\hat{r}}(\bar{x}) = \prod_{\sigma\in\mathcal{X}} B_{\hat{r}_\sigma}(\bar{x}_\sigma)$. By applying the triangle inequality we find

$$\begin{aligned} \|\pi^\sigma[T(x) - \bar{x}]\|_{\ell^1_\nu} &\le \|\pi^\sigma[T(x) - T(\bar{x})]\|_{\ell^1_\nu} + \|\pi^\sigma[T(\bar{x}) - \bar{x}]\|_{\ell^1_\nu} \\ &\le \|\pi^\sigma[T(x) - T(\bar{x})]\|_{\ell^1_\nu} + Y_\sigma, \end{aligned} \tag{2.15}$$

for any $\sigma \in \mathcal{X}$ and $x \in \mathcal{B}_{\hat{r}}(\bar{x})$. Next, we introduce $\tilde{x}_k$, which denotes that the last k variables of x are fixed, e.g. $\tilde{x}_0 = x$, $\tilde{x}_n = \bar{x}$, and $\tilde{x}_{n-3} = (x_{\rho_1}, x_{\rho_2}, x_{\rho_3}, \bar{x}_{\rho_4}, \dots, \bar{x}_{\rho_n})$. Furthermore, $\tilde{x}^j_k$ denotes that the last k variables are fixed and that the j-th variable gets a prime $'$, e.g. $\tilde{x}^2_{n-3} = (x_{\rho_1}, x'_{\rho_2}, x_{\rho_3}, \bar{x}_{\rho_4}, \dots, \bar{x}_{\rho_n})$. Using this notation, and applying the triangle inequality and the mean value theorem (Lemma 2.2) we find

$$\begin{aligned} \|\pi^\sigma[T(x) - T(\bar{x})]\|_{\ell^1_\nu} &\le \sum_{k=1}^n \|\pi^\sigma[T(\tilde{x}_{k-1}) - T(\tilde{x}_k)]\|_{\ell^1_\nu} \\ &\le \sum_{k=1}^n \sup_{x'_{\rho_k} \in B_{\hat{r}_{\rho_k}}(\bar{x}_{\rho_k})} \|D_{\rho_k}\pi^\sigma T(\tilde{x}^k_{n-k})\|_{B(\ell^1_\nu,\ell^1_\nu)} \|x'_{\rho_k} - \bar{x}_{\rho_k}\|_{\ell^1_\nu} \\ &\le \sum_{\rho\in\mathcal{X}} Z_{\sigma\rho}(\hat{r})\hat{r}_\rho. \end{aligned} \tag{2.16}$$

Combining Equations (2.15) and (2.16), we obtain

$$\|\pi^\sigma[T(x) - \bar{x}]\|_{\ell^1_\nu} \le \sum_{\rho\in\mathcal{X}} Z_{\sigma\rho}(\hat{r})\hat{r}_\rho + Y_\sigma = p_\sigma(\hat{r}) + \hat{r}_\sigma.$$

By assumption that the radii polynomial $p(r)$ is negative component-wise for some $\hat{r} \in (\mathbb{R}^+)^n$, we obtain $\|\pi^\sigma[T(x) - \bar{x}]\|_{\ell^1_\nu} < \hat{r}_\sigma$ for any $\sigma \in \mathcal{X}$. We conclude that T maps $\mathcal{B}_{\hat{r}}(\bar{x})$ into itself.

Part 2 (T is a contraction on $\mathcal{B}_r(\bar{x})$)

To prove that T is a contraction on $\mathcal{B}_r(\bar{x})$ we need to choose a norm on the product space $X = (\ell^1_\nu)^n$. We show that T contracts with respect to the weighted norm

$$\|x\|_X := \max_{\sigma \in \mathcal{X}} \left\{ \frac{\|x_\sigma\|_{\ell^1_\nu}}{\hat{r}_\sigma} \right\}. \tag{2.17}$$

Let $x_1,\ x_2 \in \mathcal{B}_{\hat{r}}(\bar{x})$, then by applying the triangle inequality and the mean value theorem (Lemma 2.2), similarly to part 1,

$$\begin{aligned}
\|\pi^\sigma[T(x_1) - T(x_2)]\|_{\ell^1_\nu} &\leq \sum_{\rho \in \mathcal{X}} \sup_{x' \in \mathcal{B}_{\hat{r}}(\bar{x})} \|D_\rho \pi^\sigma T(x')\|_{B(\ell^1_\nu, \ell^1_\nu)} \|\pi^\rho(x_1 - x_2)\|_{\ell^1_\nu} \\
&\leq \sum_{\rho \in \mathcal{X}} Z_{\sigma\rho}(\hat{r}) \hat{r}_\rho \|x_1 - x_2\|_X,
\end{aligned}$$

for any $\sigma \in \mathcal{X}$. Using the definition of the weighted norm (2.17), we find

$$\|T(x_1) - T(x_2)\|_X \leq \max_{\sigma \in \mathcal{X}} \left\{ \frac{\sum_{\rho \in \mathcal{X}} Z_{\sigma\rho}(\hat{r}) \hat{r}_\rho}{\hat{r}_\sigma} \right\} \|x_1 - x_2\|_X,$$

and by the component-wise negativity of the radii polynomial together with the assumption $Y \in (\mathbb{R}^+)^n$ we conclude that

$$\max_{\sigma \in \mathcal{X}} \left\{ \frac{\sum_{\rho \in \mathcal{X}} Z_{\sigma\rho}(\hat{r}) \hat{r}_\rho}{\hat{r}_\sigma} \right\} < 1.$$

We have thereby shown that T is a contraction mapping.

Part 3 (Apply Banach contraction mapping principle)

By part 1, we obtained that $T : \mathcal{B}_{\hat{r}}(\bar{x}) \to \mathcal{B}_{\hat{r}}(\bar{x})$, with closed subset $\mathcal{B}_{\hat{r}}(\bar{x}) \subseteq X$. By part 2, we obtained that T is a contraction. The Banach contraction mapping principle (Lemma 2.3) completes the proof. □

3. Examples

In this chapter we discuss two basic examples demonstrating the theory of Chapter 2. We first discuss an ODE system with only one variable and only one differential equation. We will use Theorem 2.1 to set up a computer-assisted proof to prove that there exists an exact, and unique solution within a ball of a certain radius around the numerical solution. Next, we consider a DAE system consisting of two variables, one differential equation and one algebraic equation. For the setup of a computer-assisted proof, we use Theorem 2.12 for product spaces. The general procedure for both examples is as follows:

1. convert the dynamical system into a root finding problem,
2. define linear operators $A^\dagger$ and A,
3. determine bounds Y and Z, and obtain the radii polynomial,
4. numerically solve the system,
5. find a radius such that the radii polynomial is negative.

For the first step we use the Taylor series expansion. We perform the second and third step for an arbitrary numerical solution to the system that we substitute later. After the fourth step, we can substitute the numerical solution that we found. Also note that in both examples we use the weighted ℓ^1-norm as presented in Definition 2.5. We refer to steps 1-3 as the mathematical part, steps 4-5 are referred to as the computational part. Lastly, we remark that we use inequality symbols '$<$', and '$>$', to denote the element-wise inequality.

3.1. ODE example

The first example we consider is an ordinary differential equation. We consider the system

$$\begin{cases} \dot{u}(t) = u(t) - (u(t))^2, \\ u(0) = u_0, \end{cases} \tag{3.1}$$

where u_0 denotes the initial value at $t = 0$.

Taylor series expansion As mentioned in the introduction of this chapter, we use the Taylor series expansion to convert the dynamical system into a root finding problem. Consider the Taylor series expansion of $u(t)$

$$u(t) = \sum_{n=0}^{\infty} a_n t^n. \tag{3.2}$$

We observe that

$$(u(t))^2 = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} a_k a_{n-k} \right) t^n, \text{ and } u'(t) = \sum_{n=0}^{\infty} (n+1) a_{n+1} t^n. \tag{3.3}$$

We also note that $u_0 = u(0) = a_0$. Substitution of expansions (3.2) and (3.3) into system (3.1) converts the ODE into an infinite set of algebraic equations F in the Taylor space

$$\begin{aligned} F_0(a) &:= a_0 - u_0 \\ F_1(a) &:= a_1 - a_0 + (a * a)_0 \\ F_2(a) &:= 2a_2 - a_1 + (a * a)_1 \\ &\vdots \\ F_n(a) &:= na_n - a_{n-1} + (a * a)_{n-1} \\ &\vdots \end{aligned}$$

where $(a * a)$ denotes the convolution product of a with itself, as defined in Definition 2.7.

Defining $\boldsymbol{A^\dagger}$ and $\boldsymbol{A}$ Analysing the expression for $F_n(a) := \boldsymbol{na_n} - a_{n-1} + (a * a)_{n-1}$, our expectation is that for large $N \in \mathbb{N}$ the bold printed part, na_n, dominates. Using this expectation we define the linear operators $A^\dagger$ and A as

$$\begin{aligned} \pi_N(A^\dagger a) &= A_N^\dagger \pi_N a, &\quad \text{and } (A^\dagger a)_n &= na_n &\quad \text{for } n > N, \\ \pi_N(Aa) &= A_N \pi_N a, &\quad \text{and } (Aa)_n &= \frac{1}{n} a_n &\quad \text{for } n > N, \end{aligned} \tag{3.4}$$

where π_N denotes the finite dimensional projection $\pi_N a = (a_0, \ldots, a_N)$. Furthermore, $A_N^\dagger := DF_N(\bar{x}_N)$ with $F_N(x_N) := \pi_N F(\iota x_N)$ as defined in Equation (2.2), and A_N is the numerical inverse of $A_N^\dagger$, i.e. $A_N \approx (A_N^\dagger)^{-1}$.

Bounds and radii polynomial Having defined linear operators $A^\dagger$ and A, we can establish the bounds (2.9) and (2.10a-c). For the bound Z we observe the following:

$$DT(\bar{a} + w)v = [I - AA^\dagger]v - A[DF(\bar{a}) - A^\dagger]v - A[DF(\bar{a} + w) - DF(\bar{a})]v, \tag{3.5}$$

where $\|v\|_{\ell^1_\nu} \leq 1$ and $w \in B_r(0)$. Note that we shifted the problem statement to the origin.

Bound $\boldsymbol{Y}$ For the first bound (2.9), we recall

$$\|AF(\bar{a})\|_{\ell^1_\nu} \leq Y.$$

Let $\bar{a} = (\bar{a}_0, \bar{a}_1, \dots, \bar{a}_N, 0, \dots)$, then

$$(F(\bar{a}))_k = \begin{cases} \bar{a}_0 - u_0 & k = 0, \\ k\bar{a}_k - \bar{a}_{k-1} + (\bar{a} * \bar{a})_{k-1} & 1 \le k \le N, \\ -\bar{a}_{k-1} + (\bar{a} * \bar{a})_{k-1} & k = N+1, \\ (\bar{a} * \bar{a})_{k-1} & N+2 \le k \le 2N, \\ 0 & k > 2N. \end{cases} \tag{3.6}$$

The Y bound becomes

$$Y := \sum_{k=0}^{N} |\,(A_N \pi_N F(\bar{a}))_k\,| \nu^k + \frac{1}{N+1} |-\bar{a}_N + (\bar{a}*\bar{a})_N| \nu^{N+1} + \sum_{k=N+2}^{2N} \frac{1}{k} |(\bar{a}*\bar{a})_{k-1}| \nu^k, \tag{3.7}$$

where we substituted (3.6) and used the definition of the ℓ^1_ν-norm.

Bound Z^0 We want to obtain a bound

$$\|I - AA^\dagger\|_{B(\ell^1_\nu, \ell^1_\nu)} \le Z^0.$$

We denote the $(N+1)$-dimensional identity matrix by I_N. Furthermore, ι denotes the embedding by extending with zeroes. Observe that, for the first term of Equation (3.5), we obtain

$$[I - AA^\dagger]v = \iota[I_N - A_N A_N^\dagger]\pi_N v,$$

based on the definition of $A^\dagger$ and A in (3.4). In other words, the first term of Equation (3.5) reduces to a finite dimensional operator. The bound Z^0 becomes

$$\|I - AA^\dagger\|_{B(\ell^1_\nu, \ell^1_\nu)} \le \max_{0 \le m \le N} \frac{1}{\nu^m} \sum_{k=0}^{N} |(I - AA^\dagger)_{km}| \nu^k =: Z^0, \tag{3.8}$$

using the upper bound on the operator norm of Proposition 2.10.

Bound Z^1 By definition of $A_N^\dagger$, as provided in (2.3), we have

$$A_N^\dagger = DF_N(\pi_N \bar{a}).$$

Furthermore, by Proposition 2.9, we obtain

$$D[(a * a)_k]v = 2(a * v)_k.$$

Using these equalities and the definition of $A^\dagger$ in (3.4), we find

$$([DF(\bar{a}) - A^\dagger]v)_k = \begin{cases} 0 & 0 \le k \le N, \\ 2(\bar{a} * v)_{k-1} - v_{k-1} & k > N. \end{cases}$$

For $k > N$, we observe

$$|(A[DF(\bar{a}) - A^\dagger]v)_k| = \frac{1}{k}|2(\bar{a} * v)_{k-1} - v_{k-1}|,$$

using the definition of A (3.4). By using the Banach algebra property of Proposition 2.8 we can bound the sum of the elements $|(A[DF(\bar{a}) - A^\dagger]v)_k|$ for $k > N$ as

$$\begin{aligned}
\sum_{k=N+1}^{\infty} |(A[DF(\bar{a}) - A^\dagger]v)_k|\nu^k &= \sum_{k=N+1}^{\infty} \frac{1}{k}|2(\bar{a} * v)_{k-1} - v_{k-1}|\nu^k \\
&\leq \frac{\nu}{N+1} \sum_{k=0}^{\infty} |2(\bar{a} * v)_k - v_k|\nu^k \\
&\leq \frac{\nu}{N+1} \left(2\sum_{k=0}^{\infty} |(\bar{a} * v)_k|\nu^k + \sum_{k=0}^{\infty} |v_k|\nu^k \right) \\
&= \frac{\nu}{N+1} \left(2\|(\bar{a} * v)\|_{\ell^1_\nu} + \|v\|_{\ell^1_\nu} \right) \\
&\leq \frac{\nu}{N+1} \left(2\|\bar{a}\|_{\ell^1_\nu}\|v\|_{\ell^1_\nu} + \|v\|_{\ell^1_\nu} \right) \\
&\leq \frac{\nu}{N+1} \left(2\|\bar{a}\|_{\ell^1_\nu} + 1 \right),
\end{aligned}$$

where we also used that $\|v\|_{\ell^1_\nu} \leq 1$ as was explicitly assumed above. We conclude that the second term of Equation (3.5) can be bounded by

$$Z^1 := \frac{\nu}{N+1} \left(2\|\bar{a}\|_{\ell^1_\nu} + 1 \right). \tag{3.9}$$

Bound $Z^2(r)$ For the final term of Equation (3.5), we note that

$$([DF(\bar{a} + w) - DF(\bar{a})]v)_k = \begin{cases} 0 & k = 0, \\ 2(w * v)_{k-1} & k \geq 1. \end{cases} \tag{3.10}$$

We first consider $0 \leq k \leq N$. Let $w = r\tilde{w}$ with $\tilde{w} \in B_1(0)$ then, with the help of the Banach algebra property (Proposition 2.8), we find

$$\|(r\tilde{w} * v)\|_{\ell^1_\nu} \leq r\|\tilde{w}\|_{\ell^1_\nu}\|v\|_{\ell^1_\nu} \leq r.$$

We obtain

$$2r\nu^{-k+1} \geq 2(r\tilde{w} * v)_{k-1}, \quad \text{for all } \|\tilde{w}\|_{\ell^1_\nu} \leq 1,\ \|v\|_{\ell^1_\nu} \leq 1,\ k \geq 1,$$

and define $C = (C_0, \ldots, C_N)$ component-wise as

$$C_k := \begin{cases} 0 & k = 0, \\ \nu^{-k} & k \geq 1. \end{cases} \tag{3.11}$$

We note that we use C_k as an upper bound for the k-th term of (3.10).

For the tail ($k > N$), we use a similar reasoning as for the bound Z^1 to obtain

$$\sum_{k=N+1}^{\infty} |(A[DF(\bar{a}+w) - DF(\bar{a})]v)_k|\nu^k \leq \frac{2\nu}{N+1}r. \tag{3.12}$$

Using C, as defined in (3.11), and (3.12) we obtain a bound $Z^2(r)$;

$$Z^2(r) := 2\nu \left(\sum_{k=0}^{N} (|A_N|C)_k \nu^k + \frac{1}{N+1} \right) r = \hat{Z}^2 r, \tag{3.13}$$

with $\hat{Z}^2 := 2\nu \left(\sum_{k=0}^{N} (|A_N|C)_k \nu^k + \frac{1}{N+1} \right)$.

Radii polynomial In line with the definition of the radii polynomial (Definition 2.4), and its alternative representation (2.11), the radii polynomial is given by

$$p(r) = Y + [Z^0 + Z^1 + \hat{Z}^2 r - 1]r, \tag{3.14}$$

where

$$\begin{aligned} Y &= \sum_{k=0}^{N} |\left(A_N \pi_N F(\bar{a})\right)_k| \nu^k + \frac{1}{N+1}| - \bar{a}_N + (\bar{a}*\bar{a})_N|\nu^{N+1} + \sum_{k=N+2}^{2N} \frac{1}{k}|(\bar{a}*\bar{a})_{k-1}|\nu^k, \\ Z^0 &= \|I - AA^\dagger\|_{B(\ell^1_\nu, \ell^1_\nu)}, \ Z^1 = \frac{\nu}{N+1}\left(2\|\bar{a}\|_{\ell^1_\nu} + 1\right), \\ \hat{Z}^2 &= 2\nu \left(\sum_{k=0}^{N} (|A_N|C)_k \nu^k + \frac{1}{N+1} \right), \end{aligned} \tag{3.15}$$

as defined in Equations (3.7) to (3.9) and (3.13)

3.1.1. Computational results

The simplicity of this ODE example allows us to analytically solve system (3.1). The general solution to the system is given by

$$u(t) = \frac{e^t}{\frac{1-u_0}{u_0} + e^t}. \tag{3.16}$$

Radius of convergence Throughout this example we used the Taylor series to transform the dynamic problem into a root-finding problem. When using power series, one must realise that a power series representation comes with a radius of convergence. The radius of convergence is the radius of the largest disk in which the power series converges.

For this simple ODE example it is easy to determine the radius of convergence. We observe that $u(z)$ of (3.16) has singularities for

$$z = \log\left(\frac{u_0 - 1}{u_0}\right),$$

where $u_0 > 0$ and $u_0 \neq 1$. For $u_0 = 1$, $u(z)$ has no singularities. The radius of convergence is then given by the distance between singularity z and the origin in the complex plane. As an example: for $u_0 = 0.5$ we find a radius of convergence of π. Since there are no singularities for $u_0 = 1$, the radius of convergence is infinite. We remark that the radius of convergence creates an upper bound on ν.

Implementation[1] Using the INTLAB package introduced Chapter 1, we implemented the bounds (3.15) and its corresponding radii polynomial (3.14) in MATLAB. Furthermore, we solved the system (3.1) using the recurrence relation

$$a_n = \frac{-a_{n-1} + (a * a)_{n-1}}{n},$$

for some initial value $a_0 = u_0$. One can also use the analytical solution (3.16) together with the built-in MATLAB function `taylor` to obtain the Taylor coefficients. We worked out both options, yielding the same results.

We ran the MATLAB script to prove that, in a ball around the N-dimensional, numerically found solution to system (3.1), an exact and unique solution exists. To illustrate the computational step, we generate a numerical solution with initial value $a_0 = 0.5$, we arbitrarily set $\nu = 0.2 < \pi$ and $N = 5$. Using this solution we test whether we can find r such that $p(r) < 0$ for varying initial values u_0. We vary the initial values to demonstrate the results we get when we start with a less accurate approximate root. The results are visualised in Figure 3.1, where we plotted the radii polynomials for initial values $u_0 = 0.6, 0.5, 0.4, 0.3$. The intervals of r such that $p(r) < 0$ are approximately given in Table 3.1.

u_0	r_{int}
0.6	$(0.1252, 0.8197)$
0.5	$(2.659 * 10^{-11}, 0.9449)$
0.4	$(0.1252, 0.8197)$
0.3	$(0.3384, 0.6065)$

Table 3.1.: Intervals returned by the MATLAB script for varying u_0.

In line with our expectation, the numerical solution generated by using $a_0 = 0.5$ lies very close to the exact solution. In fact, by the computational step we find that the exact

[1]Script used: `script1D.m`

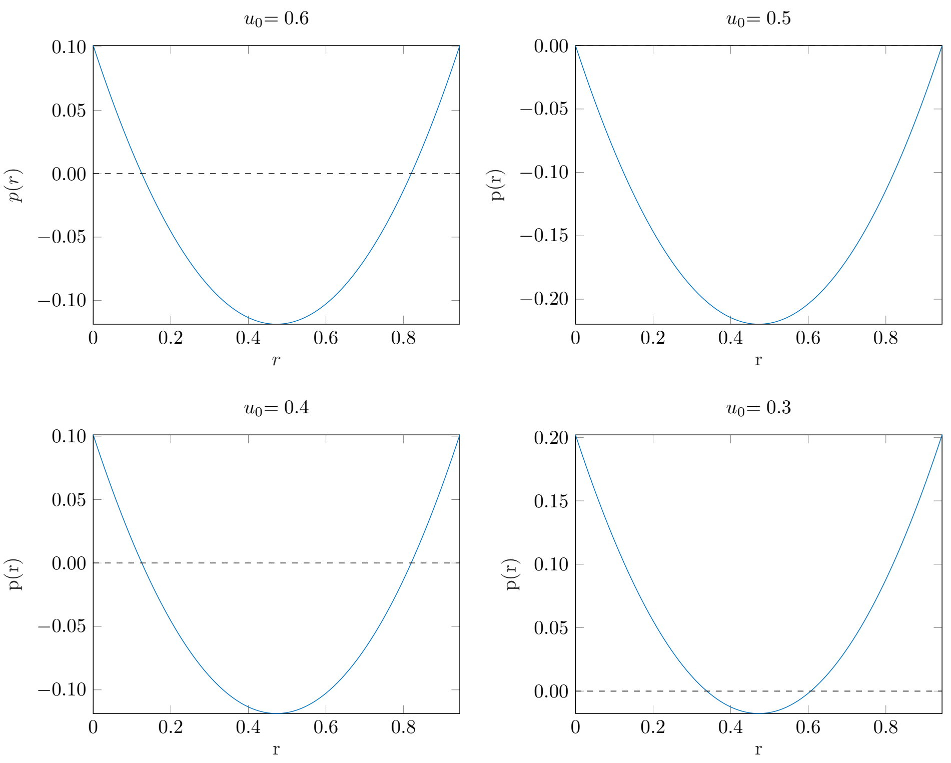


Figure 3.1.: Figure illustrating the radii polynomial p. The dashed lines represent the line $p(r) = 0$. Any value of r for which $p(r)$ lies below this line suffices for the proof that an exact and unique solution exists in a ball around the numerical solution with this specific radius r.

solution to the problem lies within a ball of radius $2.659 * 10^{-12}$ around the numerical solution. When checking whether this solution holds for a different initial value u_0, we observe that we find a much larger minimal r, as expected. Lastly, we note that we verified the numerical solution only for time interval $t \in [0, \nu] = [0, 0.2]$.

Continuation[2] If we want to verify a numerical solution on a larger time interval, we can explore numerous options. First we can try to stretch the time interval by increasing ν. However, we must note that, to be successful, we need a more precise numerical solution. We can improve accuracy of the solution by using Newton's method (Section 2.1) on F. We can also increase accuracy by calculating more Taylor coefficients, and thus increasing N. We do note that, since we are performing numeric calculations, we are limited in increasing the accuracy. Another obstacle we face is the radius of convergence, which creates an upper bound on ν. We also note that, for this example, we did not implement Newton's method since it does not improve the numerical solution (we already use F to determine the coefficients $a = (a_0, \dots, a_N)$).

If we want to further increase the length of the time interval, we can attempt the

[2]Script used: `continuation1D.m`

following

1. Find an accurate numerical solution
2. Choose a ν and validate numerical solution for $t \in [0, \nu]$
3. Find the minimal r from the interval obtained in step 2
4. Determine new starting point $a_0' = \sum_{n=0}^{N} a_n \nu^n \pm r$
5. Repeat steps 1-4 for $u_0 = a_0'$

Using the above procedure we can verify a numerical solution on a larger time interval. Note that after the first iteration, the numerical solution is no longer a vector of values but a vector of intervals.

Using the parameters

$$N = 10, \quad \nu = 0.5, \quad u_0 = 0.5,$$

we managed to verify the numerical solution on a time interval of $[0, 3.5]$, where we note that we surpassed the bound on ν created by the radius of convergence π. After seven iterations of the above procedure, the lower bound on r became too large to continue validating, i.e. the numerical solution, after seven iterations, was no longer accurate enough.

To plot the results we observe that

$$u(t) = \sum_{n=0}^{\infty} (\bar{a} + r\tilde{w})_n t^n = \sum_{n=0}^{N} \bar{a}_n t^n + r \sum_{n=0}^{\infty} \tilde{w}_n t^n,$$

with $\|\tilde{w}\|_{\ell_\nu^1} \leq 1$. The difference between the exact solution and the numerical solution on the time interval $[0, \nu]$ can be bounded as follows:

$$\left| u(t) - \sum_{n=0}^{N} \bar{a}_n t^n \right| \leq r \sum_{n=0}^{\infty} |\tilde{w}_n| t^n \leq r \sum_{n=0}^{\infty} |\tilde{w}_n| \nu^n \leq r \|\tilde{w}\|_{\ell_\nu^1} \leq r. \tag{3.17}$$

Using the bound (3.17), we can plot the area around the numerically computed solution in which the exact solution lies, for every time step. The validated solution is presented in Figure 3.2. We observe that for the first time steps the radius is too small to illustrate a difference between the shaded area and the exact solution. However, for the last successful time step $[3, 3.5]$ we observe a larger radius, which can be (partially) explained by the accumulation of computational errors and radii of every time step.

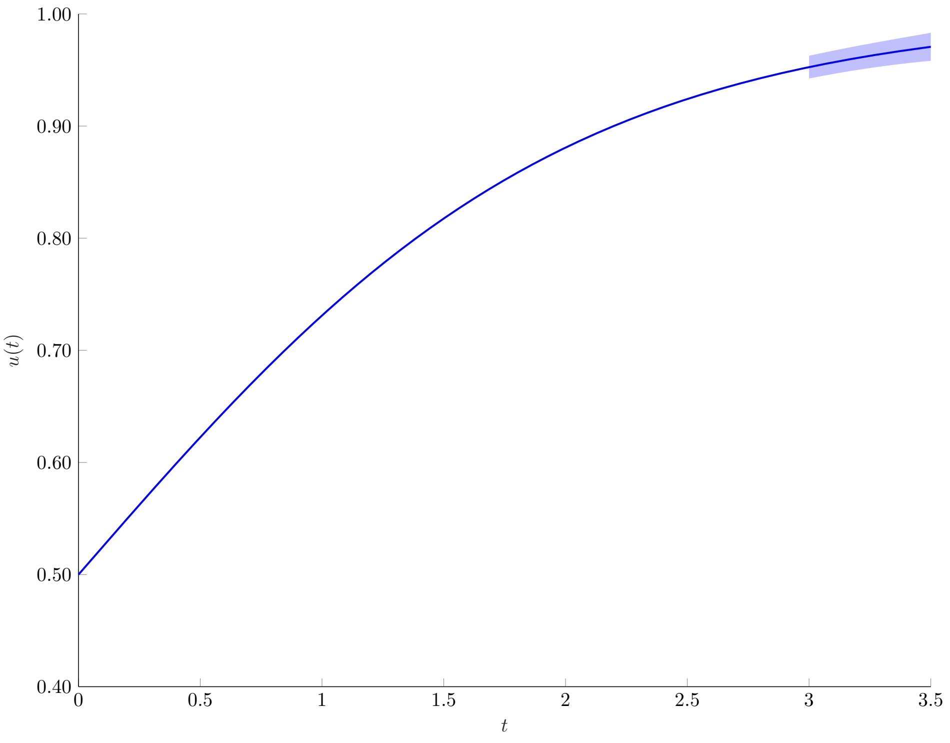


Figure 3.2.: Figure illustrating a plot of the exact solution $u(t)$ given by (3.16), in blue. The shaded area illustrates the area where the exact solution should be, based on the computed numerical guess and the smallest r that we determined.

3.2. DAE example

For the second example, we consider the following system

$$\begin{cases} \dot{u}(t) = \dfrac{u(t)}{L + u(t)}, \\ u(0) = u_0. \end{cases} \tag{3.18}$$

where u_0 denotes the initial value at $t = 0$, and L denotes some positive constant; $L \in \mathbb{R}^+$. We rewrite system (3.18) to

$$\begin{cases} \dot{u}(t) = u(t)v(t), \\ \quad 1 = (L + u(t))v(t), \\ u(0) = u_0, \end{cases} \tag{3.19}$$

where we introduced an extra algebraic equation for the denominator of the system. Observe that system (3.19) is a differential-algebraic system.

Taylor series expansion Similar to the ODE example, we use the Taylor series expansion to convert the dynamical system into a root-finding problem. Consider the Taylor series expansions of $u(t)$, $v(t)$;

$$u(t) = \sum_{n=0}^{\infty} a_n t^n, \quad v(t) = \sum_{n=0}^{\infty} b_n t^n, \tag{3.20}$$

we observe that

$$u(t)v(t) = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} a_k b_{n-k} \right) t^n, \text{ and } u'(t) = \sum_{n=0}^{\infty} (n+1) a_{n+1} t^n. \tag{3.21}$$

We also note that $u_0 = u(0) = a_0$ and that we have an analytic expression for $v(0) = \frac{1}{L+a_0}$. We decide to split off a_0, b_0 and consider these known (in fact $u_0 = a_0$ is known and we have an analytic expression for $v(0) = \frac{1}{L+a_0}$). We carry out this split in order to obtain a factor of ν in the bounds of Z^1 and $Z^2(r)$. We can thereby ensure that these bounds can be made small by decreasing ν.

We define $a_{n+1} =: \tilde{a}_n \in \ell^1_\nu$, $n \geq 0$, and similarly $b_{n+1} =: \tilde{b}_n \in \ell^1_\nu$, $n \geq 0$. Observe that splitting a_0, and b_0 off the convolution product gives

$$(a * b)_n = \begin{cases} a_0 b_0 & n = 0, \\ a_0 \tilde{b}_0 + b_0 \tilde{a}_0 & n = 1, \\ a_0 \tilde{b}_{n-1} + b_0 \tilde{a}_{n-1} + (\tilde{a} * \tilde{b})_{n-2} & n \geq 2. \end{cases}$$

By substitution of expansions (3.20) and (3.21) into system (3.19) and splitting off a_0, b_0 we convert the differential-algebraic system into two infinite sets of algebraic equations $f_{\tilde{a}}$, $f_{\tilde{b}}$ in the Taylor space

$$(f_{\tilde{a}})_n(x) := \begin{cases} \tilde{a}_0 - a_0 b_0 & n = 0, \\ 2\tilde{a}_1 - a_0 \tilde{b}_0 - b_0 \tilde{a}_0 & n = 1, \\ (n+1)\tilde{a}_n - a_0 \tilde{b}_{n-1} - b_0 \tilde{a}_{n-1} - (\tilde{a} * \tilde{b})_{n-2} & n \geq 2, \end{cases}$$

$$(f_{\tilde{b}})_n(x) := \begin{cases} (L + a_0)\tilde{b}_0 + b_0 \tilde{a}_0 & n = 0, \\ (L + a_0)\tilde{b}_n + b_0 \tilde{a}_n + (\tilde{a} * \tilde{b})_{n-1} & n \geq 1, \end{cases}$$

where $(\tilde{a} * \tilde{b})$ denotes the convolution product of $\tilde{a}$ with $\tilde{b}$, as defined in Definition 2.7. Furthermore we recall that a_0 and b_0 are not variably.

Let us now define $F(x) := (f_{\tilde{a}}(x), f_{\tilde{b}}(x))$, with $x = (\tilde{a}, \tilde{b})$, then we have obtained an equation for every unknown $\tilde{a}_i$, $\tilde{b}_i$, $i \in \mathbb{N}_0$ and have thereby converted system (3.19) into a root finding problem $F(x) = 0$ in the product space $X = \ell^1_\nu \times \ell^1_\nu$.

Throughout this example we extensively use the notation of a linear operator, as presented in Remark 3. Furthermore, we split $f_{\tilde{a}}$, and $f_{\tilde{b}}$ into two parts, see Remark 4.

Remark 3. Let Γ be a linear operator on the product space $X = \ell^1_\nu \times \ell^1_\nu$. We can decompose Γ as

$$\Gamma = \begin{bmatrix} \Gamma_{\tilde{a}\tilde{a}} & \Gamma_{\tilde{a}\tilde{b}} \\ \Gamma_{\tilde{b}\tilde{a}} & \Gamma_{\tilde{b}\tilde{b}} \end{bmatrix},$$

where $\Gamma_{\tilde{a}\tilde{a}}$, $\Gamma_{\tilde{a}\tilde{b}}$, $\Gamma_{\tilde{b}\tilde{a}}$, $\Gamma_{\tilde{b}\tilde{b}} \in \ell^1_\nu$. More formally stated,

$$\begin{aligned} \pi^{\tilde{a}}\Gamma x &= \Gamma_{\tilde{a}\tilde{a}}\pi^{\tilde{a}}x + \Gamma_{\tilde{a}\tilde{b}}\pi^{\tilde{b}}x \\ \pi^{\tilde{b}}\Gamma x &= \Gamma_{\tilde{b}\tilde{a}}\pi^{\tilde{a}}x + \Gamma_{\tilde{b}\tilde{b}}\pi^{\tilde{b}}x, \end{aligned}$$

where $\pi^{\tilde{a}}x = (\tilde{a}_0, \tilde{a}_1, \dots)$ and $\pi^{\tilde{b}}x = (\tilde{b}_0, \tilde{b}_1, \dots)$.

Remark 4. We will split every infinite vector of functions f into two parts; the N-dimensional finite part f^N and the infinite dimensional tail $\tilde{f}$. For instance, if $x = (a, b)$, then

$$f_a(x) = \begin{pmatrix} f_a^N(x) \\ \tilde{f}_a(x) \end{pmatrix}, \quad f_b(x) = \begin{pmatrix} f_b^N(x) \\ \tilde{f}_b(x) \end{pmatrix},$$

and

$$F(x) = \begin{pmatrix} f_a^N(x) \\ \tilde{f}_a(x) \\ f_b^N(x) \\ \tilde{f}_b(x) \end{pmatrix}.$$

Defining $A^\dagger$ and A In line with Remark 3, we define $A^\dagger$ and A as

$$A = \begin{pmatrix} A_{\tilde{a}\tilde{a}} & A_{\tilde{a}\tilde{b}} \\ A_{\tilde{b}\tilde{a}} & A_{\tilde{b}\tilde{b}} \end{pmatrix}, \quad A^\dagger = \begin{pmatrix} A^\dagger_{\tilde{a}\tilde{a}} & A^\dagger_{\tilde{a}\tilde{b}} \\ A^\dagger_{\tilde{b}\tilde{a}} & A^\dagger_{\tilde{b}\tilde{b}} \end{pmatrix},$$

with

$$\begin{aligned}
\pi_N^{\tilde{a}}(A^\dagger_{\tilde{a}\tilde{a}}x) &= \bar{A}^\dagger_{\tilde{a}\tilde{a}}\pi_N^{\tilde{a}}x, &&\text{and } (\lambda_{\tilde{a}\tilde{a}})_n := \pi^{\tilde{a}}(A^\dagger_{\tilde{a}\tilde{a}}x)_n = (n+1)\tilde{a}_n &&\text{for } n \ge N, \\
\pi_N^{\tilde{b}}(A^\dagger_{\tilde{a}\tilde{b}}x) &= \bar{A}^\dagger_{\tilde{a}\tilde{b}}\pi_N^{\tilde{b}}x, &&\text{and } (\lambda_{\tilde{a}\tilde{b}})_n := \pi^{\tilde{b}}(A^\dagger_{\tilde{a}\tilde{b}}x)_n = 0 &&\text{for } n \ge N, \\
\pi_N^{\tilde{a}}(A^\dagger_{\tilde{b}\tilde{a}}x) &= \bar{A}^\dagger_{\tilde{b}\tilde{a}}\pi_N^{\tilde{a}}x, &&\text{and } (\lambda_{\tilde{b}\tilde{a}})_n := \pi^{\tilde{a}}(A^\dagger_{\tilde{b}\tilde{a}}x)_n = b_0\tilde{a}_n &&\text{for } n \ge N, \\
\pi_N^{\tilde{b}}(A^\dagger_{\tilde{b}\tilde{b}}x) &= \bar{A}^\dagger_{\tilde{b}\tilde{b}}\pi_N^{\tilde{b}}x, &&\text{and } (\lambda_{\tilde{b}\tilde{b}})_n := \pi^{\tilde{b}}(A^\dagger_{\tilde{b}\tilde{b}}x)_n = (L+a_0)\tilde{b}_n &&\text{for } n \ge N.
\end{aligned}$$

where π_N denotes the finite dimensional projection

$$\pi_N x = (x_0, \dots, x_{N-1}), \quad \pi_N^{\tilde{a}}x = (\tilde{a}_0, \dots, \tilde{a}_{N-1}), \quad \pi_N^{\tilde{b}}x = (\tilde{b}_0, \dots, \tilde{b}_{N-1}),$$

and

$$\begin{aligned} \bar{A}^\dagger_{\tilde{a}\tilde{a}} &= \pi_N D_{\tilde{a}} f_{\tilde{a}}^N(\bar{x}), & \bar{A}^\dagger_{\tilde{a}\tilde{b}} &= \pi_N D_{\tilde{b}} f_{\tilde{a}}^N(\bar{x}), \\ \bar{A}^\dagger_{\tilde{b}\tilde{a}} &= \pi_N D_{\tilde{a}} f_{\tilde{b}}^N(\bar{x}), & \bar{A}^\dagger_{\tilde{b}\tilde{b}} &= \pi_N D_{\tilde{b}} f_{\tilde{b}}^N(\bar{x}). \end{aligned} \tag{3.22}$$

As in the ODE example we expect that $(n+1)\tilde{a}_n$ is the dominant part of $(f_{\tilde{a}})_n$ for n large. Furthermore, we expect the linear part of $(f_{\tilde{b}})_n$, $(L+a_0)\tilde{b}_n + b_0\tilde{a}_n$ to dominate for large n since the coefficients of a Taylor series decay geometrically and the remaining part of $(f_{\tilde{b}})_n$ contains solely products of those coefficients.

We introduce Λ to denote the infinite tails of $A^\dagger$

$$\begin{aligned} \Lambda_{\tilde{a}\tilde{a}} &:= \operatorname{diag}((\lambda_{\tilde{a}\tilde{a}})_N, (\lambda_{\tilde{a}\tilde{a}})_{N+1}, \dots), \quad \Lambda_{\tilde{a}\tilde{b}} := \operatorname{diag}((\lambda_{\tilde{a}\tilde{b}})_N, (\lambda_{\tilde{a}\tilde{b}})_{N+1}, \dots), \\ \Lambda_{\tilde{b}\tilde{a}} &:= \operatorname{diag}((\lambda_{\tilde{b}\tilde{a}})_N, (\lambda_{\tilde{b}\tilde{a}})_{N+1}, \dots), \quad \Lambda_{\tilde{b}\tilde{b}} := \operatorname{diag}((\lambda_{\tilde{b}\tilde{b}})_N, (\lambda_{\tilde{b}\tilde{b}})_{N+1}, \dots). \end{aligned}$$

We conclude that we have $A^\dagger$ of the form

$$A^\dagger = \left[\begin{array}{cc|cc} \bar{A}^\dagger_{\tilde{a}\tilde{a}} & 0 & \bar{A}^\dagger_{\tilde{a}\tilde{b}} & 0 \\ 0 & \Lambda_{\tilde{a}\tilde{a}} & 0 & 0 \\ \hline \bar{A}^\dagger_{\tilde{b}\tilde{a}} & 0 & \bar{A}^\dagger_{\tilde{b}\tilde{b}} & 0 \\ 0 & \Lambda_{\tilde{b}\tilde{a}} & 0 & \Lambda_{\tilde{b}\tilde{b}} \end{array}\right],$$

where by 0 we denote the matrix filled with zeros of the appropriate size. We define A as its approximate inverse

$$A = \left[\begin{array}{cc|cc} \bar{A}_{\tilde{a}\tilde{a}} & 0 & \bar{A}_{\tilde{a}\tilde{b}} & 0 \\ 0 & \Lambda^{-1}_{\tilde{a}\tilde{a}} & 0 & 0 \\ \hline \bar{A}_{\tilde{b}\tilde{a}} & 0 & \bar{A}_{\tilde{b}\tilde{b}} & 0 \\ 0 & \Lambda^{-1}_{\tilde{b}\tilde{a}} & 0 & \Lambda^{-1}_{\tilde{b}\tilde{b}} \end{array}\right],$$

where we numerically invert $\bar{A}^\dagger$ to obtain $\bar{A}$, or more precisely $(\bar{A}^\dagger)^{-1} \approx \bar{A}$ with

$$\begin{aligned} \bar{A}_{\tilde{a}\tilde{a}} &= -(\bar{A}^\dagger_{\tilde{b}\tilde{a}})^{-1}\bar{A}^\dagger_{\tilde{b}\tilde{b}}(\bar{A}^\dagger_{\tilde{a}\tilde{b}} - \bar{A}^\dagger_{\tilde{a}\tilde{a}}(\bar{A}^\dagger_{\tilde{b}\tilde{a}})^{-1}\bar{A}^\dagger_{\tilde{b}\tilde{b}})^{-1}, \quad \bar{A}_{\tilde{a}\tilde{b}} = (\bar{A}^\dagger_{\tilde{b}\tilde{a}} - \bar{A}^\dagger_{\tilde{b}\tilde{b}}(\bar{A}^\dagger_{\tilde{a}\tilde{b}})^{-1}\bar{A}^\dagger_{\tilde{a}\tilde{a}})^{-1}, \\ \bar{A}_{\tilde{b}\tilde{a}} &= (\bar{A}^\dagger_{\tilde{a}\tilde{b}} - \bar{A}^\dagger_{\tilde{a}\tilde{a}}(\bar{A}^\dagger_{\tilde{b}\tilde{a}})^{-1}\bar{A}^\dagger_{\tilde{b}\tilde{b}})^{-1}, \quad \bar{A}_{\tilde{b}\tilde{b}} = -(\bar{A}^\dagger_{\tilde{a}\tilde{b}})^{-1}\bar{A}^\dagger_{\tilde{a}\tilde{a}}(\bar{A}^\dagger_{\tilde{b}\tilde{a}} - \bar{A}^\dagger_{\tilde{b}\tilde{b}}(\bar{A}^\dagger_{\tilde{a}\tilde{b}})^{-1}\bar{A}^\dagger_{\tilde{a}\tilde{a}})^{-1}. \end{aligned}$$

Inverting Λ yields

$$\begin{aligned} \Lambda^{-1}_{\tilde{a}\tilde{a}} &= \operatorname{diag}((\lambda^{-1}_{\tilde{a}\tilde{a}})_N, (\lambda^{-1}_{\tilde{a}\tilde{a}})_{N+1}, \dots), \\ \Lambda^{-1}_{\tilde{a}\tilde{b}} &= \operatorname{diag}(0, 0, \dots), \\ \Lambda^{-1}_{\tilde{b}\tilde{a}} &= \operatorname{diag}(-(\lambda_{\tilde{b}\tilde{a}})_N(\lambda^{-1}_{\tilde{a}\tilde{a}})_N(\lambda^{-1}_{\tilde{b}\tilde{b}})_N, -(\lambda_{\tilde{b}\tilde{a}})_{N+1}(\lambda^{-1}_{\tilde{a}\tilde{a}})_{N+1}(\lambda^{-1}_{\tilde{b}\tilde{b}})_{N+1}, \dots), \\ \Lambda^{-1}_{\tilde{b}\tilde{b}} &= \operatorname{diag}((\lambda^{-1}_{\tilde{b}\tilde{b}})_N, (\lambda^{-1}_{\tilde{b}\tilde{b}})_{N+1}, \dots). \end{aligned}$$

Note that inverting an infinite dimensional matrix is far from trivial. However, due to the specific form of Λ inverting analytically is possible. Key to this successful analytic inverse is that Λ is composed of diagonal block matrices.

Bounds and radii polynomial Following Theorem 2.12 of Section 2.6, we need to find bounds in the product space $X = \ell^1_\nu \times \ell^1_\nu$ on the Fréchet differentiable map $T(x) = x - AF(x)$ with $x \in X$. In this example, $\mathcal{X} = (\tilde{a}, \tilde{b})$ denotes the ordered list of variables. Recall that we need to find some scalars $Y_\sigma \in \mathbb{R}^+$, and functions $Z_{\sigma\rho} : \mathbb{R}^+ \to \mathbb{R}^+$, with σ, $\rho \in \mathcal{X}$ such that for every $r = (r_{\tilde{a}}, r_{\tilde{b}}) \in (\mathbb{R}^+)^2$,

$$\|\pi^\sigma[T(\bar{x}) - \bar{x}]\|_{\ell^1_\nu} \leq Y_\sigma \qquad \text{for all } \sigma \in \mathcal{X}, \tag{2.13}$$

$$\|D_\rho \pi^\sigma T(x)\|_{B(\ell^1_\nu, \ell^1_\nu)} \leq Z_{\sigma\rho}(r) \quad \text{for all } x \in \mathcal{B}_r(\bar{x}),\ \sigma, \rho \in \mathcal{X}. \tag{2.14}$$

Observe that we need to find two bounds $Y_{\tilde{a}}$, $Y_{\tilde{b}}$ and four bounds $Z_{\tilde{a}\tilde{a}}(r)$, $Z_{\tilde{a}\tilde{b}}(r)$, $Z_{\tilde{b}\tilde{a}}(r)$, $Z_{\tilde{b}\tilde{b}}(r)$. For every $Z_{\sigma\rho}(r)$ we use a similar split as presented in Section 2.4 and worked out in the ODE example, i.e.

$$Z_{\sigma\rho}(r) = Z^0_{\sigma\rho} + Z^1_{\sigma\rho} + Z^2_{\sigma\rho}(r), \quad \sigma\rho \in \mathcal{X}. \tag{3.23}$$

Lastly, we use $v = (v_{\tilde{a}}, v_{\tilde{b}})$, with $\|v_{\tilde{a}}\|_{\ell^1_\nu} \leq 1$, $\|v_{\tilde{b}}\|_{\ell^1_\nu} \leq 1$ to determine the bounds Z^0, Z^1, $Z^2(r)$.

Bound *Y* For the first bound (2.13), we want to determine $Y_{\tilde{a}}$, $Y_{\tilde{b}}$ such that

$$\|\pi^{\tilde{a}} AF(\bar{x})\|_{\ell^1_\nu} \leq Y_{\tilde{a}},$$
$$\|\pi^{\tilde{b}} AF(\bar{x})\|_{\ell^1_\nu} \leq Y_{\tilde{b}}.$$

Let $\bar{\tilde{a}} = (\bar{\tilde{a}}_0, \ldots, \bar{\tilde{a}}_{N-1}, 0, \ldots)$, $\bar{\tilde{b}} = \left(\bar{\tilde{b}}_0, \ldots, \bar{\tilde{b}}_{N-1}, 0, \ldots\right)$, and $\bar{x} = (\bar{\tilde{a}}, \bar{\tilde{b}})$. Using Remark 3 we obtain

$$\pi^{\tilde{a}} AF(\bar{x}) = A_{\tilde{a}\tilde{a}} f_{\tilde{a}}(\bar{x}) + A_{\tilde{a}\tilde{b}} f_{\tilde{b}}(\bar{x}),$$
$$\pi^{\tilde{b}} AF(\bar{x}) = A_{\tilde{b}\tilde{a}} f_{\tilde{a}}(\bar{x}) + A_{\tilde{b}\tilde{b}} f_{\tilde{b}}(\bar{x}).$$

Next we observe that

$$F(\bar{x}) = \begin{pmatrix} f_{\tilde{a}}(\bar{x}) \\ f_{\tilde{b}}(\bar{x}) \end{pmatrix},$$

with

$$(f_{\tilde{a}}(\bar{x}))_n = \begin{cases} \bar{\tilde{a}}_0 - a_0 b_0 & n = 0, \\ 2\bar{\tilde{a}}_1 - a_0 \bar{\tilde{b}}_0 - b_0 \bar{\tilde{a}}_0 & n = 1, \\ (n+1)\bar{\tilde{a}}_n - a_0 \bar{\tilde{b}}_{n-1} - b_0 \bar{\tilde{a}}_{n-1} - (\bar{\tilde{a}} * \bar{\tilde{b}})_{n-2} & 1 < n < N, \\ -\, a_0 \bar{\tilde{b}}_{n-1} - b_0 \bar{\tilde{a}}_{n-1} - (\bar{\tilde{a}} * \bar{\tilde{b}})_{n-2} & n = N, \\ -\, (\bar{\tilde{a}} * \bar{\tilde{b}})_{n-2} & N < n \leq 2N, \\ 0 & n > 2N, \end{cases}$$

$$(f_{\tilde{b}}(\bar{x}))_n = \begin{cases} (L + a_0)\bar{\tilde{b}}_0 + b_0 \bar{\tilde{a}}_0 & n = 0, \\ (L + a_0)\bar{\tilde{b}}_n + b_0 \bar{\tilde{a}}_n + (\bar{\tilde{a}} * \bar{\tilde{b}})_{n-1} & 0 < n < N, \\ (\bar{\tilde{a}} * \bar{\tilde{b}})_{n-1} & N \leq n < 2N, \\ 0 & n \geq 2N. \end{cases}$$

The Y bound becomes

$$\begin{aligned}
Y_{\tilde{a}} &= \sum_{n=0}^{N-1} |(\bar{A}_{\tilde{a}\tilde{a}} f_{\tilde{a}}^N(\bar{x}))_n|\nu^n + \sum_{n=0}^{N-1} |(\bar{A}_{\tilde{a}\tilde{b}} f_{\tilde{b}}^N(\bar{x}))_n|\nu^n + \sum_{n=N}^{2N} \frac{1}{n+1}|(f_{\tilde{a}}(\bar{x}))_n|\nu^n, \\
Y_{\tilde{b}} &= \sum_{n=0}^{N-1} |(A_{\tilde{b}\tilde{a}} f_{\tilde{a}}^N(\bar{x}))_n|\nu^n + \sum_{n=0}^{N-1} |(A_{\tilde{b}\tilde{b}} f_{\tilde{b}}^N(\bar{x}))_n|\nu^n \\
&\quad + \sum_{n=N}^{2N} \frac{b_0}{(n+1)(L+a_0)}|(f_{\tilde{a}}(\bar{x}))_n|\nu^n + \sum_{n=N}^{2N-1} \frac{1}{L+a_0}|(f_{\tilde{b}}(\bar{x}))_n|\nu^n.
\end{aligned} \tag{3.24}$$

Bound Z^0 Let $v = \begin{pmatrix} v_{\tilde{a}} \\ v_{\tilde{b}} \end{pmatrix}$. We observe

$$[I - AA^\dagger]v = \begin{pmatrix} \iota[I_N - \bar{A}_{\tilde{a}\tilde{a}}\bar{A}^\dagger_{\tilde{a}\tilde{a}} - \bar{A}_{\tilde{a}\tilde{b}}\bar{A}^\dagger_{\tilde{b}\tilde{a}}] & \iota[-\bar{A}_{\tilde{a}\tilde{b}}\bar{A}^\dagger_{\tilde{a}\tilde{a}} - \bar{A}_{\tilde{a}\tilde{b}}\bar{A}^\dagger_{\tilde{b}\tilde{b}}] \\ \iota[-\bar{A}_{\tilde{b}\tilde{a}}\bar{A}^\dagger_{\tilde{a}\tilde{a}} - \bar{A}_{\tilde{b}\tilde{b}}\bar{A}^\dagger_{\tilde{a}\tilde{b}}] & \iota[I_N - \bar{A}_{b\tilde{a}}\bar{A}^\dagger_{\tilde{a}\tilde{b}} - \bar{A}_{\tilde{b}\tilde{b}}\bar{A}^\dagger_{\tilde{b}\tilde{b}}] \end{pmatrix} \begin{pmatrix} v_{\tilde{a}} \\ v_{\tilde{b}} \end{pmatrix},$$

where I_N denotes the N-dimensional identity matrix, and ι denotes the embedding by extending with zeroes. Write $\Gamma = I - AA^\dagger$, then using Remark 3,

$$\begin{aligned}
\pi^{\tilde{a}}\Gamma x &= \Gamma_{\tilde{a}\tilde{a}}\pi^{\tilde{a}}x + \Gamma_{\tilde{a}\tilde{b}}\pi^{\tilde{b}}x \\
\pi^{\tilde{b}}\Gamma x &= \Gamma_{\tilde{b}\tilde{a}}\pi^{\tilde{a}}x + \Gamma_{\tilde{b}\tilde{b}}\pi^{\tilde{b}}x.
\end{aligned}$$

As in the ODE example, we observe that the four linear operators, $\Gamma_{\tilde{a}\tilde{a}}$, $\Gamma_{\tilde{a}\tilde{b}}$, $\Gamma_{\tilde{b}\tilde{a}}$, $\Gamma_{\tilde{b}\tilde{b}}$, reduce to finite N-dimensional operators. Thus, we obtain

$$\begin{aligned}
\|\Gamma_{\tilde{a}\tilde{a}}\|_{B(\ell^1_\nu,\ell^1_\nu)} &\le \max_{0\le m\le N-1} \frac{1}{\nu^m}\sum_{k=0}^{N-1} |(\Gamma_{\tilde{a}\tilde{a}})_{km}|\nu^k =: Z^0_{\tilde{a}\tilde{a}}, \\
\|\Gamma_{\tilde{a}\tilde{b}}\|_{B(\ell^1_\nu,\ell^1_\nu)} &\le \max_{0\le m\le N-1} \frac{1}{\nu^m}\sum_{k=0}^{N-1} |(\Gamma_{\tilde{a}\tilde{b}})_{km}|\nu^k =: Z^0_{\tilde{a}\tilde{b}}, \\
\|\Gamma_{\tilde{b}\tilde{a}}\|_{B(\ell^1_\nu,\ell^1_\nu)} &\le \max_{0\le m\le N-1} \frac{1}{\nu^m}\sum_{k=0}^{N-1} |(\Gamma_{\tilde{b}\tilde{a}})_{km}|\nu^k =: Z^0_{\tilde{b}\tilde{a}}, \\
\|\Gamma_{\tilde{b}\tilde{b}}\|_{B(\ell^1_\nu,\ell^1_\nu)} &\le \max_{0\le m\le N-1} \frac{1}{\nu^m}\sum_{k=0}^{N-1} |(\Gamma_{\tilde{b}\tilde{b}})_{km}|\nu^k =: Z^0_{\tilde{b}\tilde{b}},
\end{aligned} \tag{3.25}$$

using the upper bound on the operator norm of Proposition 2.10.

Bound Z^1 We observe that

$$DF(x) = \left[\begin{array}{cccccc|ccccc}
1 & 0 & \dots & \dots & \dots & \dots & 0 & \dots & \dots & \dots & \dots \\
-b_0 & 2 & 0 & \dots & \dots & \dots & -a_0 & 0 & \dots & \dots & \dots \\
-\tilde{b}_0 & -b_0 & 3 & 0 & \dots & \dots & -\tilde{a}_0 & -a_0 & 0 & \dots & \dots \\
-\tilde{b}_1 & -\tilde{b}_0 & -b_0 & 4 & 0 & \dots & -\tilde{a}_1 & -\tilde{a}_0 & -a_0 & 0 & \dots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \ddots & \ddots & \ddots & \ddots \\
\hline
b_0 & 0 & \dots & \dots & \dots & \dots & L+a_0 & 0 & \dots & \dots & \dots \\
\tilde{b}_0 & b_0 & 0 & \dots & \dots & \dots & \tilde{a}_0 & L+a_0 & 0 & \dots & \dots \\
\tilde{b}_1 & \tilde{b}_0 & b_0 & 0 & \dots & \dots & \tilde{a}_1 & \tilde{a}_0 & L+a_0 & 0 & \dots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \ddots & \ddots & \ddots & \ddots
\end{array}\right],$$

and obtain

$$\left(\left(Df_{\tilde{a}}(\bar{x}) - \begin{bmatrix} A^\dagger_{\tilde{a}\tilde{a}} & A^\dagger_{\tilde{a}\tilde{b}} \end{bmatrix}\right) v\right)_n = \begin{cases} 0 & 0 \le n < N, \\ -\,(\bar{\tilde{b}} * v_{\tilde{a}})_{n-2} - (\bar{\tilde{a}} * v_{\tilde{b}})_{n-2} & \\ \quad -\,a_0 (v_{\tilde{b}})_{n-1} - b_0 (v_{\tilde{a}})_{n-1} & n \ge N, \end{cases} \tag{3.26}$$

$$\left(\left(Df_{\tilde{b}}(\bar{x}) - \begin{bmatrix} A^\dagger_{\tilde{b}\tilde{a}} & A^\dagger_{\tilde{b}\tilde{b}} \end{bmatrix}\right) v\right)_n = \begin{cases} 0 & 0 \le n < N, \\ (\bar{\tilde{b}} * v_{\tilde{a}})_{n-1} + (\bar{\tilde{a}} * v_{\tilde{b}})_{n-1} & n \ge N, \end{cases} \tag{3.27}$$

using the definition of $A^\dagger$. Note that we constructed the tails of $A^\dagger$ in such a way that we subtract the diagonals of the four blocks of $DF(\bar{x})$. Furthermore, as in the ODE example, the first N rows of Equations (3.26) and (3.27) are zero because of the construction of $\bar{A}^\dagger$ in (3.22). We note that,

$$[DF(\bar{x}) - A^\dagger]v = \begin{pmatrix} \left(Df_{\tilde{a}}(\bar{x}) - \begin{bmatrix} A^\dagger_{\tilde{a}\tilde{a}} & A^\dagger_{\tilde{a}\tilde{b}} \end{bmatrix}\right) v \\ \left(Df_{\tilde{b}}(\bar{x}) - \begin{bmatrix} A^\dagger_{\tilde{b}\tilde{a}} & A^\dagger_{\tilde{b}\tilde{b}} \end{bmatrix}\right) v \end{pmatrix},$$

and for $n \ge N$ we observe

$$\left(\begin{bmatrix} A_{\tilde{a}\tilde{a}} & A_{\tilde{a}\tilde{b}} \end{bmatrix} [DF(\bar{x}) - A^\dagger] v\right)_n = -\frac{(\bar{\tilde{b}} * v_{\tilde{a}})_{n-2} + (\bar{\tilde{a}} * v_{\tilde{b}})_{n-2} + a_0 (v_{\tilde{b}})_{n-1} + b_0 (v_{\tilde{a}})_{n-1}}{n+1},$$

and

$$\begin{aligned}\left(\begin{bmatrix} A_{\tilde{b}\tilde{a}} & A_{\tilde{b}\tilde{b}} \end{bmatrix} [DF(\bar{x}) - A^\dagger] v\right)_n = {} & \frac{b_0 \left((\bar{\tilde{b}} * v_{\tilde{a}})_{n-2} + (\bar{\tilde{a}} * v_{\tilde{b}})_{n-2} + a_0 (v_{\tilde{b}})_{n-1} + b_0 (v_{\tilde{a}})_{n-1}\right)}{(n+1)(L+a_0)} \\ & + \frac{(\bar{\tilde{b}} * v_{\tilde{a}})_{n-1} + (\bar{\tilde{a}} * v_{\tilde{b}})_{n-1}}{L+a_0}.\end{aligned}$$

Using the same arguments as in the ODE example we find for Z^1,

$$
\begin{aligned}
Z^1_{\tilde{a}\tilde{a}} &:= \frac{\nu^2\|\bar{\tilde{b}}\|_{\ell^1_\nu} + \nu b_0}{N+1}, \\
Z^1_{\tilde{a}\tilde{b}} &:= \frac{\nu^2\|\bar{\tilde{a}}\|_{\ell^1_\nu} + \nu a_0}{N+1}, \\
Z^1_{\tilde{b}\tilde{a}} &:= \frac{b_0\left(\nu^2\|\bar{\tilde{b}}\|_{\ell^1_\nu} + \nu b_0\right)}{(N+1)(L+a_0)} + \frac{\nu\|\bar{\tilde{b}}\|_{\ell^1_\nu}}{L+a_0}, \\
Z^1_{\tilde{b}\tilde{b}} &:= \frac{b_0\left(\nu^2\|\bar{\tilde{a}}\|_{\ell^1_\nu} + \nu a_0\right)}{(N+1)(L+a_0)} + \frac{\nu\|\bar{\tilde{a}}\|_{\ell^1_\nu}}{L+a_0}.
\end{aligned}
\tag{3.28}
$$

Bound $Z^2(r)$ Let $w = \begin{pmatrix} r_{\tilde{a}}\tilde{w}_{\tilde{a}} \\ r_{\tilde{b}}\tilde{w}_{\tilde{b}} \end{pmatrix}$, $r = \begin{pmatrix} r_{\tilde{a}} \\ r_{\tilde{b}} \end{pmatrix}$, with $\tilde{w}_{\tilde{a}}$, $\tilde{w}_{\tilde{b}} \in B_1(0)$, then

$$
([Df_{\tilde{a}}(\bar{x}+w) - Df_{\tilde{a}}(\bar{x})]v)_n = \begin{cases} 0 & \text{for } 0 \le n \le 1, \\ -r_{\tilde{b}}(\tilde{w}_{\tilde{b}} * v_{\tilde{a}})_{n-2} - r_{\tilde{a}}(\tilde{w}_{\tilde{a}} * v_{\tilde{b}})_{n-2} & \text{for } n > 1, \end{cases}
$$

and

$$
([Df_{\tilde{b}}(\bar{x}+w) - Df_{\tilde{b}}(\bar{x})]v)_n = \begin{cases} 0 & \text{for } n = 0, \\ r_{\tilde{b}}(\tilde{w}_{\tilde{b}} * v_{\tilde{a}})_{n-1} + r_{\tilde{a}}(\tilde{w}_{\tilde{a}} * v_{\tilde{b}})_{n-1} & \text{for } n > 0. \end{cases}
$$

Then, using the same arguments as in the ODE example we define

$$
C_{\tilde{a}} := (0, 0, \nu^{-2}, \nu^{-3}, \dots, \nu^{-N+1}), \quad C_{\tilde{b}} := (0, \nu^{-1}, \nu^{-2}, \dots, \nu^{-N+1}).
$$

The bounds Z^2 can then be determined as

$$
\begin{aligned}
Z^2_{\tilde{a}\tilde{a}}(r) &= \left(\nu\||\bar{A}_{\tilde{a}\tilde{a}}|C_{\tilde{a}}\|_{\ell^1_\nu} + \||\bar{A}_{\tilde{a}\tilde{b}}|C_{\tilde{b}}\|_{\ell^1_\nu} + \frac{\nu}{N+1}\right)\nu r_{\tilde{b}}, \\
Z^2_{\tilde{a}\tilde{b}}(r) &= \left(\nu\||\bar{A}_{\tilde{a}\tilde{a}}|C_{\tilde{a}}\|_{\ell^1_\nu} + \||\bar{A}_{\tilde{a}\tilde{b}}|C_{\tilde{b}}\|_{\ell^1_\nu} + \frac{\nu}{N+1}\right)\nu r_{\tilde{a}}, \\
Z^2_{\tilde{b}\tilde{a}}(r) &= \left(\nu\||\bar{A}_{\tilde{b}\tilde{a}}|C_{\tilde{a}}\|_{\ell^1_\nu} + \||\bar{A}_{\tilde{b}\tilde{b}}|C_{\tilde{b}}\|_{\ell^1_\nu} + \frac{b_0\nu}{(N+1)(L+a_0)} + \frac{1}{L+a_0}\right)\nu r_{\tilde{b}}, \\
Z^2_{\tilde{b}\tilde{b}}(r) &= \left(\nu\||\bar{A}_{\tilde{b}\tilde{a}}|C_{\tilde{a}}\|_{\ell^1_\nu} + \||\bar{A}_{\tilde{b}\tilde{b}}|C_{\tilde{b}}\|_{\ell^1_\nu} + \frac{b_0\nu}{(N+1)(L+a_0)} + \frac{1}{L+a_0}\right)\nu r_{\tilde{a}},
\end{aligned}
\tag{3.29}
$$

where, for readability, we use the norm notation instead of the sum over the N-dimensional vectors.

Radii polynomial In line with the definition of the multi dimensional radii polynomial (Definition 2.11), and extending the alternative representation of (2.11), as presented in (3.23), the radii polynomial is given by

$$
p(r) = Y + [Z^0 + Z^1 + Z^2(r) - I]r, \tag{3.30}
$$

where I denotes the two-dimensional identity matrix and

$$Y = \begin{pmatrix} Y_{\tilde{a}} \\ Y_{\tilde{b}} \end{pmatrix}, \quad Z^i = \begin{pmatrix} Z^i_{\tilde{a}\tilde{a}} & Z^i_{\tilde{a}\tilde{b}} \\ Z^i_{\tilde{b}\tilde{a}} & Z^i_{\tilde{b}\tilde{b}} \end{pmatrix}, \quad i = 0, 1, 2, \tag{3.31}$$

given by Equations (3.24), (3.25), (3.28) and (3.29).

3.2.1. Computational results

Solving system (3.18) analytically is much harder than in the ODE example. To obtain Taylor coefficients a_n, b_n we can either use built-in MATLAB solvers, e.g. `ode15i`, or use the recurrence relations. Since we do not have an analytical solution we cannot determine singularities of the solution. It is less simple to acquire radii of convergence for the power series of u and v.

Finding radius vector r satisfying $p(r) < 0$ is also more complex. In the ODE example, where radii polynomial p consisted of only one polynomial and depended on only one radius, it was rather easy to find all r satisfying $p(r) < 0$. In this example, finding all $r = (r_{\tilde{a}}, r_{\tilde{b}})$ satisfying $p(r) = (p_{\tilde{a}}(r), p_{\tilde{b}}(r)) < 0$ is not so trivial. One can visualise this challenge as finding all points r such that the two (intersecting) graphs $p_{\tilde{a}}(r)$, $p_{\tilde{b}}(r)$ are negative.

Implementation[3] As in the ODE example we use the INTLAB package, introduced in Chapter 1, to implement the bounds (3.31) and the two-dimensional radii polynomial (3.30). We are particularly interested in finding the 'smallest' r such that all conditions are satisfied. Since r is a two-dimensional vector of radii, we need to define what we mean by smallest. We want to find the pair $(r_{\tilde{a}}, r_{\tilde{b}})$ that lies closest to the origin $(0, 0)$, in the L_1-distance (also known as the Manhatten distance), and satisfies $p(r) < 0$. We choose to work with the L_1-distance since we want to find an r such that the difference between the components of r is small, proving useful in the continuation explained in the next paragraph.

We introduce the function $\tilde{p}(r)$:

$$\tilde{p}(r) := \begin{cases} K r_{\text{bnd}}(r_{\tilde{a}} + r_{\tilde{b}}) & p(r) < 0, \\ 2K r_{\text{bnd}} + \max\limits_{\sigma \in \{a,b\}} p_\sigma(r) & \text{otherwise}, \end{cases} \tag{3.32}$$

where r_{bnd} is a predetermined upper bound on $r_{\tilde{a}}$ and $r_{\tilde{b}}$, K is a scaling factor to increase numerical accuracy, and the value 2 represents the number of variables of the system. We set $K = 10^6$ and $r_{\text{bnd}} = 1$. If we now minimise $\tilde{p}(r)$ for r, we hope to find the r closest to the origin, i.e. we hope to find the global minimum of $\tilde{p}(r)$.

We ran the computation for parameters

$$N = 5, \quad \nu = 0.3, \quad u_0 = 0.5, \quad L = 0.3.$$

[3]Scripts used: `images_ex2D.m`, `script2D.m`

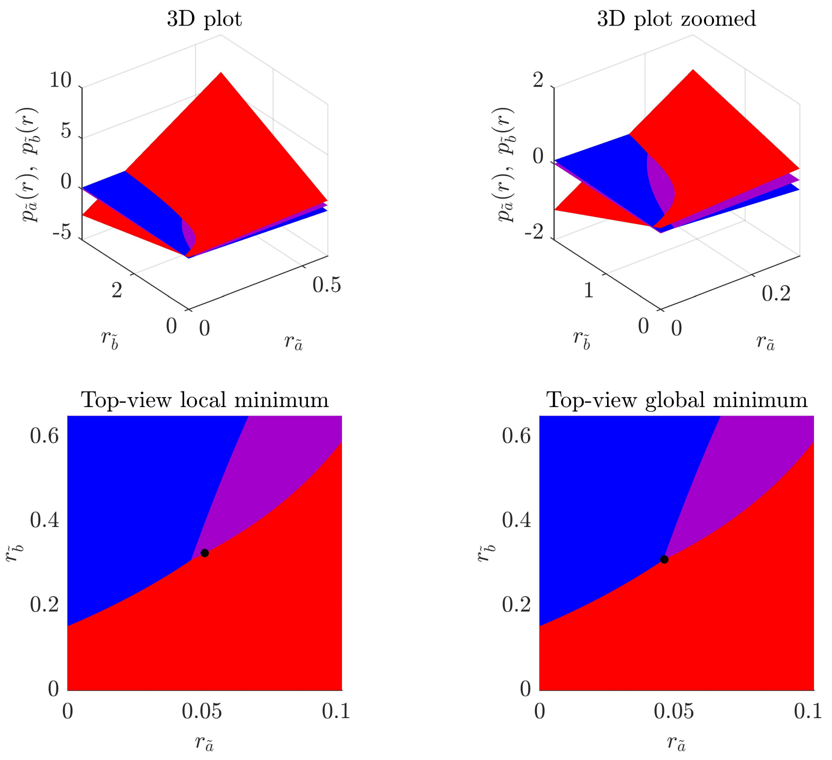


Figure 3.3.: This plot visualises the two radii polynomials $p_{\tilde{a}}(r)$ and $p_{\tilde{b}}(r)$, in blue and red, respectively. The purple plane is the plane $z = 0$. In the right-hand plot a top view of this 3D plot is visualised. The black marker denotes the pair $r = (r_{\tilde{a}}, r_{\tilde{b}})$ such that $p(r) < 0$ found by our script illustrating the local, or global, minimum found. The purple area in the top-view plots illustrates possible combinations of $(r_{\tilde{a}}, r_{\tilde{b}})$ such that both polynomials are negative. For illustration purposes we deliberately used an inaccurate numerical approximation, explaining the large values of $(r_{\tilde{a}}, r_{\tilde{b}})$.

The results are visualised in Figure 3.3. By setting the flag `glob` to true, the script attempts to find a global minimiser for $\tilde{p}(r)$ and checks rigorously that the componentwise negativity of the radii polynomial holds for this numerically found global minimiser. Searching for a global minimiser is obviously more time consuming than finding an arbitrary r satisfying $p(r) < 0$.

Continuation[4] Following an equivalent approach as in the ODE example, one can verify a numerical solution on a larger time interval. The standard built-in solvers of MATLAB do not produce an accurate enough result for the continuation to work. The solution produced by the solvers are even not accurate enough for the multivariate Newton method to converge. To overcome this challenge we use the recurrence relations

[4]Script used: `continuation2D.m`

directly to determine a numerical solution to system (3.18).

As an example, we ran the script with parameters

$$N = 20, \quad \nu = 0.35, \quad u_0 = 0.5, \quad L = 0.3,$$

and managed to verify the numerical solution for $t \in [0, 2.8]$. We observe that, as the number of time steps increases, the inaccuracy of the numerical solution increases, and thereby also the vector r that we find increases. As in the ODE example, we can plot the validated solution using the upper bound (3.17) for both $u(t)$ and $v(t)$. The validated solution is visualised in Figure 3.4. In this figure we observe indeed that the area in which the exact solution lies increases as time progresses.

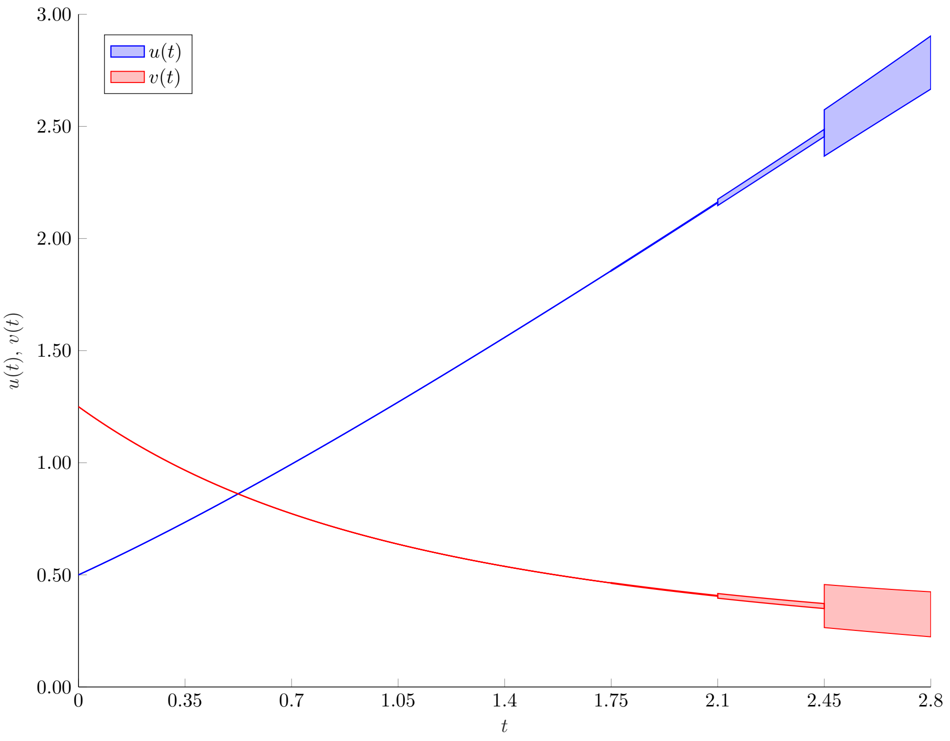


Figure 3.4.: Figure illustrating the area where the exact solution should be, based on the computed numerical guess and the smallest $r_{\tilde{a}}$, $r_{\tilde{b}}$ that we determined.

4. Analytical results

In this chapter, we discuss the simplified qORAC system (1.5) introduced in Chapter 1. We recall that the goal is to find balls, around a numerically computed solution to the system, that contain an exact, and unique solution. To achieve this goal we follow a similar approach as in Chapter 3, where we discussed both an ODE and a DAE example. The example in Section 3.2 proves particularly useful since similar challenges need to be overcome in this chapter.

4.1. Simplified qORAC

Recall that we have the system

$$\begin{cases} \dot{u} = e_1 f_1(\underline{v}, u) - e_2 f_2(u), \\ \dot{e}_1 = E_1(\phi, u) - e_1, \\ \dot{e}_2 = E_2(\phi, u) - e_2, \\ 0 = \dfrac{\partial O}{\partial u}(\phi, u), \end{cases} \tag{1.5}$$

with

$$\begin{aligned} f_1(\underline{v}, u) &= \frac{\underline{v} - K_{\text{eq}} u}{K_1 + u + \underline{v}}, & f_2(u) &= \frac{u}{L + u}, \\ E_1(\phi, u) &= \frac{1}{f_1(\phi, u)} \frac{1}{O(\phi, u)}, & E_2(\phi, u) &= 1 - E_1(\phi, u), \\ O(\phi, u) &= \frac{1}{f_1(\phi, u)} + \frac{1}{f_2(u)}, & K_1,\ K_{\text{eq}},\ L,\ \underline{v} &\in \mathbb{R}^+. \end{aligned} \tag{4.1}$$

As in Section 3.2, we rewrite the differential equations of this system. We eliminate the fractions in the differential equations by adding algebraic equations, and obtain

$$\begin{cases} \dot{u} = g\left(e_1(\underline{v} - K_{\text{eq}} u)(L + u) - e_2 u (K_1 + u + \underline{v})\right), \\ \dot{e}_1 = (K_1 + u + \phi) u h - e_1, \\ \dot{e}_2 = 1 - (K_1 + u + \phi) u h - e_2, \\ 1 = g (K_1 + u + \underline{v})(L + u), \\ 1 = h\left((K_1 + u + \phi) u + (L + u)(\phi - K_{\text{eq}} u)\right), \\ 0 = (1 + K_{\text{eq}}) \phi u^2 + (K_1 K_{\text{eq}} - L K_{\text{eq}}^2) u^2 - L \phi^2 + 2 L K_{\text{eq}} \phi u. \end{cases} \tag{4.2}$$

where we also substituted Equations (4.1).

4.1.1. Taylor Series Expansion

To obtain an infinite set of equations, and thereby convert system (4.2) into a root-finding problem, we again substitute the Taylor series expansion for functions u, e_1, e_2, g, h, ϕ; i.e.

$$\begin{pmatrix} u(t) \\ e_1(t) \\ e_2(t) \\ g(t) \\ h(t) \\ \phi(t) \end{pmatrix} = \begin{pmatrix} \sum_{n=0}^{\infty} a_n t^n \\ \sum_{n=0}^{\infty} b_n t^n \\ \sum_{n=0}^{\infty} c_n t^n \\ \sum_{n=0}^{\infty} \alpha_n t^n \\ \sum_{n=0}^{\infty} \beta_n t^n \\ \sum_{n=0}^{\infty} \gamma_n t^n \end{pmatrix}. \tag{4.3}$$

Note that, to improve clarity, we use the latin alphabet to denote the coefficients of Taylor series for function of the differential variables of the system (u, e_1, e_2). For the algebraic variables (g, h, ϕ) we use the greek alphabet. As in the DAE example, we face issues finding proper, small bounds Y, Z^0, Z^1, $Z^2(r)$. To overcome this, we split off the initial values. Note that this does not create any issues in the differential part of the system, since we have an exact expression for initial values $u(0) = a_0$, $e_1(0) = b_0$, and $e_2(0) = c_0$. For the algebraic part, we obtain the following expressions

$$\alpha_0 = \frac{1}{(K_1 + \underline{v})L + (K_1 + \underline{v} + L)a_0 + a_0^2}, \tag{4.4a}$$

$$\beta_0 = \frac{1}{(K_1 - LK_{\text{eq}})a_0 + (1 - K_{\text{eq}})a_0^2 + 2a_0\gamma_0 + L\gamma_0}, \tag{4.4b}$$

$$\gamma_0^{\pm} = \frac{(1 + K_{\text{eq}})a_0^2 + 2LK_{\text{eq}}a_0 \pm \sqrt{(1 + K_{\text{eq}})^2 a_0^4 + 4LK_{\text{eq}}(1 + K_{\text{eq}})a_0^3 + 4K_1K_{\text{eq}}La_0^2}}{2L}. \tag{4.4c}$$

Observe that there are two options for γ_0 in (4.4c), throughout the rest of this chapter, we use γ_0 to refer to either one of them. In Chapter 5, we decide to work with γ_0^+. We also note that α_0, β_0, $\gamma_0 \in \mathbb{R}$ if a_0, b_0, $c_0 \in \mathbb{R}^+$, where we used K_1, K_{eq}, L, $\underline{v} \in \mathbb{R}^+$. To conclude this subsection, we define $a_{n+1} =: \tilde{a}_n \in \ell_\nu^1$, $n \geq 0$, and in a similar manner define $\tilde{b}_n$, $\tilde{c}_n$, $\tilde{\alpha}_n$, $\tilde{\beta}_n$, and $\tilde{\gamma}_n$. Using this notation we obtain an expression for the infinite set of equations

$$F(x) = \begin{pmatrix} f_{\tilde{a}}(x) \\ f_{\tilde{b}}(x) \\ f_{\tilde{c}}(x) \\ f_{\tilde{\alpha}}(x) \\ f_{\tilde{\beta}}(x) \\ f_{\tilde{\gamma}}(x)) \end{pmatrix}, \quad \text{with } x = (\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}).$$

The functions f_σ, $\sigma \in \{\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}\}$ are given by

$$\begin{aligned}
(f_{\tilde{a}}(x))_n &= (n+1)\tilde{a}_n - \underline{v}L(b * \alpha)_n - (\underline{v} - LK_{\mathrm{eq}})(a * b * \alpha)_n + K_{\mathrm{eq}}(a * a * b * \alpha)_n \\
&\qquad + (K_1 + \underline{v})(a * c * \alpha)_n + (a * a * c * \alpha)_n, \\
(f_{\tilde{b}}(x))_n &= (n+1)\tilde{b}_n + b_n - K_1(a * \beta)_n - (a * a * \beta)_n - (a * \beta * \gamma)_n, \\
(f_{\tilde{c}}(x))_n &= (n+1)\tilde{c}_n + c_n - (\mathcal{I})_n + K_1(a * \beta)_n + (a * a * \beta)_n + (a * \beta * \gamma)_n, \\
(f_{\tilde{\alpha}}(x))_n &= (K_1 + \underline{v})L\tilde{\alpha}_n + (K_1 + \underline{v} + L)(a * \alpha)_{n+1} + (a * a * \alpha)_{n+1}, \\
(f_{\tilde{\beta}}(x))_n &= (K_1 - LK_{\mathrm{eq}})(a * \beta)_{n+1} + L(\beta * \gamma)_{n+1} + 2(a * \beta * \gamma)_{n+1} \\
&\qquad + (1 - K_{\mathrm{eq}})(a * a * \beta)_{n+1}, \\
(f_{\tilde{\gamma}}(x))_n &= (1 + K_{\mathrm{eq}})(a * a * \gamma)_{n+1} + (K_1 K_{\mathrm{eq}} - LK_{\mathrm{eq}}^2)(a * a)_{n+1} - L(\gamma * \gamma)_{n+1} \\
&\qquad + 2LK_{\mathrm{eq}}(a * \gamma)_{n+1},
\end{aligned}$$

where $*$ denotes the convolution product, as defined in Definition 2.7, and $\mathcal{I}$ is defined by

$$(\mathcal{I})_n := \begin{cases} 1 & n = 0, \\ 0 & n > 0. \end{cases}$$

Note that we did not substitute the tilde notation for the convolution products, since this would result in many extra terms notation-wise.

Remark 5. Throughout this chapter we use the notation introduced in Remarks 3 and 4, extended to the Taylor product space $X = (\ell^1_\nu)^6$. Furthermore, we use $\mathcal{X}$ to denote the ordered list of variables as in Section 2.6, i.e. $\mathcal{X} = (\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$.

4.1.2. Defining $A^\dagger$ and A

As explained in Section 2.2, we need to define an injective operator A that approximates the inverse of the Jacobian $DF(\bar{x})$, where $\bar{x} := \iota\bar{x}_N$ denotes an approximate root of $F(x)$. This operator A must be balanced in simplicity and accuracy. In line with the examples of Chapter 3, we split the definition of $A^\dagger$ into two pieces; the numerical part $\bar{A}^\dagger$, and the tails Λ. The structure of $A^\dagger$ is as follows

$$A^\dagger = \begin{bmatrix}
\bar{A}^\dagger_{\tilde{a}\tilde{a}} & 0 & \bar{A}^\dagger_{\tilde{a}\tilde{b}} & 0 & \bar{A}^\dagger_{\tilde{a}\tilde{c}} & 0 & \bar{A}^\dagger_{\tilde{a}\tilde{\alpha}} & 0 & 0 & 0 & 0 & 0 \\
0 & \Lambda_{\tilde{a}\tilde{a}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\bar{A}^\dagger_{\tilde{b}\tilde{a}} & 0 & \bar{A}^\dagger_{\tilde{b}\tilde{b}} & 0 & 0 & 0 & 0 & 0 & \bar{A}^\dagger_{\tilde{b}\tilde{\beta}} & 0 & \bar{A}^\dagger_{\tilde{b}\tilde{\gamma}} & 0 \\
0 & 0 & 0 & \Lambda_{\tilde{b}\tilde{b}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\bar{A}^\dagger_{\tilde{c}\tilde{a}} & 0 & 0 & 0 & \bar{A}^\dagger_{\tilde{c}\tilde{c}} & 0 & 0 & 0 & \bar{A}^\dagger_{\tilde{c}\tilde{\beta}} & 0 & \bar{A}^\dagger_{\tilde{c}\tilde{\gamma}} & 0 \\
0 & 0 & 0 & 0 & 0 & \Lambda_{\tilde{c}\tilde{c}} & 0 & 0 & 0 & 0 & 0 & 0 \\
\bar{A}^\dagger_{\tilde{\alpha}\tilde{a}} & 0 & 0 & 0 & 0 & 0 & \bar{A}^\dagger_{\tilde{\alpha}\tilde{\alpha}} & 0 & 0 & 0 & 0 & 0 \\
0 & \Lambda_{\tilde{\alpha}\tilde{a}} & 0 & 0 & 0 & 0 & 0 & \Lambda_{\tilde{\alpha}\tilde{\alpha}} & 0 & 0 & 0 & 0 \\
\bar{A}^\dagger_{\tilde{\beta}\tilde{a}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{A}^\dagger_{\tilde{\beta}\tilde{\beta}} & 0 & \bar{A}^\dagger_{\tilde{\beta}\tilde{\gamma}} & 0 \\
0 & \Lambda_{\tilde{\beta}\tilde{a}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Lambda_{\tilde{\beta}\tilde{\beta}} & 0 & \Lambda_{\tilde{\beta}\tilde{\gamma}} \\
\bar{A}^\dagger_{\tilde{\gamma}\tilde{a}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{A}^\dagger_{\tilde{\gamma}\tilde{\gamma}} & 0 \\
0 & \Lambda_{\tilde{\gamma}\tilde{a}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Lambda_{\tilde{\gamma}\tilde{\gamma}}
\end{bmatrix}, \qquad (4.5)$$

where by 0 we denote the matrix filled with zeroes of the appropriate size. This structure allows us to treat $\bar{A}^\dagger$ and Λ separately. The finite, numerical part $\bar{A}^\dagger$ is constructed as in (3.22) extended to six variables. For the tails Λ and its inverse Λ^{-1} we use the following notation

$$\Lambda_{\sigma\rho} = \operatorname{diag}((\lambda_{\sigma\rho})_N, (\lambda_{\sigma\rho})_{N+1}, \dots), \quad \Lambda^{-1}_{\sigma\rho} = \operatorname{diag}((\lambda^{-1}_{\sigma\rho})_N, (\lambda^{-1}_{\sigma\rho})_{N+1}, \dots).$$

where $\sigma, \rho \in \mathcal{X}$. The λ's are then defined as

$$\begin{aligned}
&(\lambda_{\tilde{a}\tilde{a}})_n = (\lambda_{\tilde{b}\tilde{b}})_n = (\lambda_{\tilde{c}\tilde{c}})_n := n+1,\\
&(\lambda_{\tilde{\alpha}\tilde{\alpha}})_n := (K_1 + \underline{v})L + (K_1 + \underline{v} + L)a_0 + a_0^2,\\
&(\lambda_{\tilde{\beta}\tilde{\beta}})_n := (K_1 - LK_{\mathrm{eq}})a_0 + (1 - K_{\mathrm{eq}})a_0^2 + 2a_0\gamma_0 + L\gamma_0,\\
&(\lambda_{\tilde{\gamma}\tilde{\gamma}})_n := (1 + K_{\mathrm{eq}})a_0^2 - 2L\gamma_0 + 2LK_{\mathrm{eq}}a_0,\\
&(\lambda_{\tilde{\alpha}\tilde{a}})_n := (2a_0 + (K_1 + \underline{v} + L))\alpha_0,\\
&(\lambda_{\tilde{\beta}\tilde{a}})_n := ((K_1 - LK_{\mathrm{eq}}) + 2\gamma_0 + 2(1 - K_{\mathrm{eq}})a_0)\beta_0,\\
&(\lambda_{\tilde{\beta}\tilde{\gamma}})_n := (L + 2a_0)\beta_0,\\
&(\lambda_{\tilde{\gamma}\tilde{a}})_n := 2(1 + K_{\mathrm{eq}})a_0\gamma_0 + 2(K_1K_{\mathrm{eq}} - LK_{\mathrm{eq}}^2)a_0 + 2LK_{\mathrm{eq}}\gamma_0.
\end{aligned}$$

Note that only $(\lambda_{\tilde{a}\tilde{a}})_n$, $(\lambda_{\tilde{b}\tilde{b}})_n$, and $(\lambda_{\tilde{c}\tilde{c}})_n$ are n-dependent. Furthermore, we can analytically invert Λ to obtain Λ^{-1}. The resulting λ^{-1}'s are given by

$$\begin{aligned}
&(\lambda^{-1}_{\tilde{a}\tilde{a}})_n = (\lambda^{-1}_{\tilde{b}\tilde{b}})_n = (\lambda^{-1}_{\tilde{c}\tilde{c}})_n := \frac{1}{n+1},\\
&(\lambda^{-1}_{\tilde{\alpha}\tilde{\alpha}})_n := \frac{1}{(\lambda_{\tilde{\alpha}\tilde{\alpha}})_n}, \quad (\lambda^{-1}_{\tilde{\beta}\tilde{\beta}})_n := \frac{1}{(\lambda_{\tilde{\beta}\tilde{\beta}})_n}, \quad (\lambda^{-1}_{\tilde{\gamma}\tilde{\gamma}})_n := \frac{1}{(\lambda_{\tilde{\gamma}\tilde{\gamma}})_n},\\
&(\lambda^{-1}_{\tilde{\alpha}\tilde{a}})_n := -\frac{(\lambda_{\tilde{\alpha}\tilde{a}})_n}{(n+1)(\lambda_{\tilde{\alpha}\tilde{\alpha}})_n}, \quad (\lambda^{-1}_{\tilde{\beta}\tilde{a}})_n := -\frac{(\lambda_{\tilde{\beta}\tilde{a}})_n(\lambda_{\tilde{\gamma}\tilde{\gamma}})_n - (\lambda_{\tilde{\beta}\tilde{\gamma}})_n(\lambda_{\tilde{\gamma}\tilde{a}})_n}{(n+1)(\lambda_{\tilde{\beta}\tilde{\beta}})_n(\lambda_{\tilde{\gamma}\tilde{\gamma}})_n},\\
&(\lambda^{-1}_{\tilde{\beta}\tilde{\gamma}})_n := -\frac{(\lambda_{\tilde{\beta}\tilde{\gamma}})_n}{(\lambda_{\tilde{\beta}\tilde{\beta}})_n(\lambda_{\tilde{\gamma}\tilde{\gamma}})_n}, \quad (\lambda^{-1}_{\tilde{\gamma}\tilde{a}})_n := -\frac{(\lambda_{\tilde{\gamma}\tilde{a}})_n}{(n+1)(\lambda_{\tilde{\gamma}\tilde{\gamma}})_n}.
\end{aligned} \tag{4.6}$$

To determine A, we first observe that it has a similar form as $A^\dagger$ as presented in (4.5). We obtain A by combining the approximate inverse $(\bar{A}^\dagger)^{-1}$ with the inverse Λ^{-1}, similar to how we combined $\bar{A}^\dagger$ and Λ in (4.5).

4.1.3. Bounds and radii polynomial

In line with Theorem 2.12 of Section 2.6, we need to find bounds in the product space $X = (\ell^1_\nu)^6$ on the Fréchet differentiable map $T(x) = x - AF(x)$ with $x \in X$. As mentioned in Remark 5, the ordered list of variables is given by $\mathcal{X} = (\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$. Recall that we need to find some scalars $Y_\sigma \in \mathbb{R}^+$, and functions $Z_{\sigma\rho} : \mathbb{R}^+ \to \mathbb{R}^+$, with σ, $\rho \in \mathcal{X}$ such that for every $r = (r_{\tilde{a}}, r_{\tilde{b}}, r_{\tilde{c}}, r_{\tilde{\alpha}}, r_{\tilde{\beta}}, r_{\tilde{\gamma}}) \in (\mathbb{R}^+)^6$,

$$\|\pi^\sigma[T(\bar{x}) - \bar{x}]\|_{\ell^1_\nu} \le Y_\sigma \qquad \text{for all } \sigma \in \mathcal{X}, \tag{2.13}$$

$$\|D_\rho \pi^\sigma T(x)\|_{B(\ell^1_\nu, \ell^1_\nu)} \le Z_{\sigma\rho}(r) \qquad \text{for all } x \in \mathcal{B}_r(\bar{x}),\ \sigma, \rho \in \mathcal{X}. \tag{2.14}$$

As in the DAE example we use, for every $Z_{\rho\sigma}(r)$, a similar split as presented in Section 2.4 and worked out in the ODE example, i.e.

$$Z_{\rho\sigma}(r) = Z^0_{\rho\sigma} + Z^1_{\rho\sigma} + Z^2_{\rho\sigma}(r). \tag{3.23}$$

To determine the bounds $Z_{\rho\sigma}(r)$, we make use of $v = (v_{\tilde{a}}, v_{\tilde{b}}, v_{\tilde{c}}, v_{\tilde{\alpha}}, v_{\tilde{\beta}}, v_{\tilde{\gamma}})$, with $\|v_\sigma\|_{\ell^1_\nu} \leq 1$, $\sigma \in \mathcal{X}$.

Bound $\boldsymbol{Y}$ We obtain a general expression for Y_σ, $\sigma \in \mathcal{X}$ by using the expressions for $\bar{A}$ and Λ^{-1} obtained in Subsection 4.1.2, and using the notation introduced in Remarks 3 and 4, extended to the Taylor product space $X = (\ell^1_\nu)^6$. Let $\sigma \in \mathcal{X}$, then

$$Y_\sigma := \sum_{\rho\in\mathcal{X}} \sum_{k=0}^{N-1} |(\bar{A}_{\sigma\rho} f^N_\rho(\bar{x}))_k| \nu^k + \sum_{\rho\in\mathcal{X}} \sum_{k=0}^{3N} |(\Lambda^{-1}_{\sigma\rho} \tilde{f}_\rho(\bar{x}))_k| \nu^{k+N}. \tag{4.7}$$

Bound $\boldsymbol{Z^0}$ Following the same procedure as in the examples of Chapter 3, we first define Γ by

$$\Gamma := I - AA^\dagger.$$

Next we observe that for all σ, $\rho \in \mathcal{X}$, $\Gamma_{\sigma\rho}$ reduces to a finite N-dimensional operator. We obtain the generalised expression for bound $Z^0_{\sigma\rho}$ using the upper bound on the operator norm of Proposition 2.10:

$$\|\Gamma_{\sigma\rho}\|_{B(\ell^1_\nu, \ell^1_\nu)} \leq \max_{0\leq m\leq N-1} \frac{1}{\nu^m} \sum_{k=0}^{N-1} |(\Gamma_{\sigma\rho})_{km}| \nu^k =: Z^0_{\sigma\rho}. \tag{4.8}$$

Bound $\boldsymbol{Z^1}$ Let σ, $\rho \in \mathcal{X}$. We need to find

$$\sum_{n=0}^{\infty} \left| \left(\sum_{d\in\mathcal{X}} A_{\sigma d}(D_\rho f_d(\bar{x}) - A^\dagger_{d\rho}) v_\rho \right)_n \right| \nu^n \leq Z^1_{\sigma\rho}.$$

We observe that for every, σ, $\rho \in \mathcal{X}$ and $0 \leq n < N$

$$((D_\rho f_\sigma(\bar{x}) - A^\dagger_{\sigma\rho}) v_\rho)_n = 0,$$

by the construction of $A^\dagger$. It remains to obtain a bound on the tails of Z^1. To obtain this bound we use the same arguments as in the ODE and DAE example. We do note that this system is more complex than the DAE example of the previous chapter. To obtain a compact notation for this bound, we note that we can bound the tails term by term. In other words, we can bound every convolution product separately.

Notation For ease of notation, we observe the following

$$
\begin{aligned}
([D_{\tilde a}(a*b)_n]v_{\tilde a})(\bar x) &= b_0(v_{\tilde a})_{n-1} + (\bar{\tilde b} * v_{\tilde a})_{n-2},\\
([D_{\tilde a}(a*b*c)_n]v_{\tilde a})(\bar x) &= b_0c_0(v_{\tilde a})_{n-1} + c_0(\bar{\tilde b} * v_{\tilde a})_{n-2} + b_0(\bar{\tilde c} * v_{\tilde a})_{n-2} + (\bar{\tilde b} * \bar{\tilde c} * v_{\tilde a})_{n-3},\\
([D_{\tilde a}(a*b*c*d)_n]v_{\tilde a})(\bar x) &= b_0c_0d_0(v_{\tilde a})_{n-1} + c_0d_0(\bar{\tilde b} * v_{\tilde a})_{n-2} + b_0d_0(\bar{\tilde c} * v_{\tilde a})_{n-2}\\
&\quad + b_0c_0(\bar{\tilde d} * v_{\tilde a})_{n-2} + d_0(\bar{\tilde b} * \bar{\tilde c} * v_{\tilde a})_{n-3} + c_0(\bar{\tilde b} * \bar{\tilde d} * v_{\tilde a})_{n-3}\\
&\quad + b_0(\bar{\tilde c} * \bar{\tilde d} * v_{\tilde a})_{n-3} + (\bar{\tilde b} * \bar{\tilde c} * \bar{\tilde d} * v_{\tilde a})_{n-4},
\end{aligned}
$$

where we determined the derivative of the convolution products. Following the exact same procedure as in the examples of Chapter 3, we bound the sum over all elements of the derivative of the convolution products by $\mathfrak{B}_b$, $\mathfrak{B}_{bc}$, $\mathfrak{B}_{bcd}$. These bounds are defined, for $\|v_{\tilde a}\|_{\ell^1_\nu} \le 1$, as

$$
\begin{aligned}
\sum_{n=0}^{\infty} |([D_{\tilde a}(a*b)_n]v_{\tilde a})(\bar x)|\nu^n &\le |b_0|\nu + \|\bar{\tilde b}\|_{\ell^1_\nu}\nu^2\\
&=: \mathfrak{B}_b,\\
\sum_{n=0}^{\infty} |([D_{\tilde a}(a*b*c)_n]v_{\tilde a})(\bar x)|\nu^n &\le |b_0c_0|\nu + (|c_0|\|\bar{\tilde b}\|_{\ell^1_\nu} + |b_0|\|\bar{\tilde c}\|_{\ell^1_\nu})\nu^2 + \|\bar{\tilde b}\|_{\ell^1_\nu}\|\bar{\tilde c}\|_{\ell^1_\nu}\nu^3\\
&=: \mathfrak{B}_{bc},\\
\sum_{n=0}^{\infty} |([D_{\tilde a}(a*b*c*d)_n]]v_{\tilde a})(\bar x)|\nu^n &\le |b_0c_0d_0|\nu + \Big((|c_0d_0|\|\bar{\tilde b}\|_{\ell^1_\nu} + |b_0d_0|\|\bar{\tilde c}\|_{\ell^1_\nu}\\
&\quad + |b_0c_0|\|\bar{\tilde d}\|_{\ell^1_\nu}\Big)\nu^2 + \Big(|d_0|\|\bar{\tilde b}\|_{\ell^1_\nu}\|\bar{\tilde c}\|_{\ell^1_\nu} + |c_0|\|\bar{\tilde b}\|_{\ell^1_\nu}\|\bar{\tilde d}\|_{\ell^1_\nu}\\
&\quad + |b_0|\|\bar{\tilde c}\|_{\ell^1_\nu}\|\bar{\tilde d}\|_{\ell^1_\nu}\Big)\nu^3 + \|\bar{\tilde b}\|_{\ell^1_\nu}\|\bar{\tilde c}\|_{\ell^1_\nu}\|\bar{\tilde d}\|_{\ell^1_\nu}\nu^4\\
&=: \mathfrak{B}_{bcd}.
\end{aligned}
$$

Lastly, we observe

$$
\begin{aligned}
\sum_{n=0}^{\infty} |([D_{\tilde a}(a*b)_{n+1}]v_{\tilde a})(\bar x)|\nu^n &\le \frac{1}{\nu}\mathfrak{B}_b,\\
\sum_{n=0}^{\infty} |([D_{\tilde a}(a*b*c)_{n+1}]v_{\tilde a})(\bar x)|\nu^n &\le \frac{1}{\nu}\mathfrak{B}_{bc},\\
\sum_{n=0}^{\infty} |([D_{\tilde a}(a*b*c*d)_{n+1}]]v_{\tilde a})(\bar x)|\nu^n &\le \frac{1}{\nu}\mathfrak{B}_{bcd}.
\end{aligned}
$$

Final bound Z^1 Using the introduced notation we can bound the tails. The final bound Z^1 becomes

$$
Z^1 := |\hat A|\hat Z^1, \tag{4.9}
$$

with

$$\hat{A} := \begin{bmatrix} (\lambda^{-1}_{\tilde{a}\tilde{a}})_N & 0 & 0 & 0 & 0 & 0 \\ 0 & (\lambda^{-1}_{\tilde{b}\tilde{b}})_N & 0 & 0 & 0 & 0 \\ 0 & 0 & (\lambda^{-1}_{\tilde{c}\tilde{c}})_N & 0 & 0 & 0 \\ (\lambda^{-1}_{\tilde{a}\tilde{\alpha}})_N & 0 & 0 & (\lambda^{-1}_{\tilde{\alpha}\tilde{\alpha}})_N & 0 & 0 \\ (\lambda^{-1}_{\tilde{a}\tilde{\beta}})_N & 0 & 0 & 0 & (\lambda^{-1}_{\tilde{\beta}\tilde{\beta}})_N & (\lambda^{-1}_{\tilde{\beta}\tilde{\gamma}})_N \\ (\lambda^{-1}_{\tilde{a}\tilde{\gamma}})_N & 0 & 0 & 0 & 0 & (\lambda^{-1}_{\tilde{\gamma}\tilde{\gamma}})_N \end{bmatrix}, \tag{4.10}$$

using the λ^{-1}'s as defined in (4.6). The formulas for $\hat{Z}^1$ are presented in Table A.1 of the Appendix A.1.

Bound $\boldsymbol{Z^2(r)}$ Let σ, $\rho \in \mathcal{X}$. We need to find

$$\sum_{n=0}^{\infty} \left| \left(\sum_{d \in \mathcal{X}} A_{\sigma d}(D_\rho f_d(\bar{x}+w) - D_\rho f_d(\bar{x}))v_\rho \right)_n \right| \nu^n \leq Z^2_{\sigma\rho}(r),$$

where

$$w = \begin{pmatrix} r_{\tilde{a}}\tilde{w}_{\tilde{a}} \\ r_{\tilde{b}}\tilde{w}_{\tilde{b}} \\ r_{\tilde{c}}\tilde{w}_{\tilde{c}} \\ r_{\tilde{\alpha}}\tilde{w}_{\tilde{\alpha}} \\ r_{\tilde{\beta}}\tilde{w}_{\tilde{\beta}} \\ r_{\tilde{\gamma}}\tilde{w}_{\tilde{\gamma}} \end{pmatrix},$$

with $\tilde{w}_\sigma \in B_1(0)$, $\sigma \in \mathcal{X}$. As in the bound Z^1, we treat the bound $Z^2(r)$ term by term (we treat the convolution products separately). To be able to bound every term, whilst maintaining a compact notation, we introduce $\mathfrak{C}$ and $\mathfrak{D}$.

Notation We first observe that

$$\begin{aligned} ([D_{\tilde{a}}(a*b)_k]v_{\tilde{a}})(\bar{x}+w) - ([D_{\tilde{a}}(a*b)_k]v_{\tilde{a}})(\bar{x}) &= (\tilde{w}_{\tilde{b}} * v_{\tilde{a}})_{k-2}r_{\tilde{b}} \\ ([D_{\tilde{a}}(a*b*c)_k]v_{\tilde{a}})(\bar{x}+w) - ([D_{\tilde{a}}(a*b*c)_k]v_{\tilde{a}})(\bar{x}), &= c_0(\tilde{w}_{\tilde{b}} * v_{\tilde{a}})_{k-2}r_{\tilde{b}} + b_0(\tilde{w}_{\tilde{c}} * v_{\tilde{a}})_{k-2}r_{\tilde{c}} \\ &\quad + (\bar{\tilde{b}} * \tilde{w}_{\tilde{c}} * v_{\tilde{a}})_{k-3}r_{\tilde{c}} \\ &\quad + (\bar{\tilde{c}} * \tilde{w}_{\tilde{b}} * v_{\tilde{a}})_{k-3}r_{\tilde{b}} \\ &\quad + (\tilde{w}_{\tilde{b}} * \tilde{w}_{\tilde{c}} * v_{\tilde{a}})_{k-3}r_{\tilde{b}}r_{\tilde{c}}. \end{aligned}$$

We omit the analogous formula for $([D_{\tilde{a}}(a*b*c*d)_k]v_{\tilde{a}})(\bar{x}+w) - ([D_{\tilde{a}}(a*b*c*d)_k]v_{\tilde{a}})(\bar{x})$ here to increase readability.

In a similar fashion as for $\mathfrak{B}$, we define $\tilde{\mathfrak{D}}$ as

$$\begin{aligned}
\tilde{\mathfrak{D}}_b &:= r_{\tilde{b}}\nu^2,\\
\tilde{\mathfrak{D}}_{bc} &:= (|c_0|r_{\tilde{b}} + |b_0|r_{\tilde{c}})\nu^2 + \left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{b}} + r_{\tilde{b}}r_{\tilde{c}}\right)\nu^3,\\
\tilde{\mathfrak{D}}_{bcd} &:= (|c_0d_0|r_{\tilde{b}} + |b_0d_0|r_{\tilde{c}} + |b_0c_0|r_{\tilde{d}})\nu^2\\
&\quad + \Bigg(|d_0|\left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{b}} + r_{\tilde{b}}r_{\tilde{c}}\right) + |c_0|\left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{b}} + r_{\tilde{b}}r_{\tilde{d}}\right)\\
&\qquad + |b_0|\left(\|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{c}} + r_{\tilde{c}}r_{\tilde{d}}\right)\Bigg)\nu^3\\
&\quad + \Bigg(\|\bar{\tilde{b}}\|_{\ell^1_\nu}\|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{b}}\|_{\ell^1_\nu}\|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu}\|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{b}}\\
&\qquad + \|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{c}}r_{\tilde{d}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{b}}r_{\tilde{d}} + \|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{b}}r_{\tilde{c}} + r_{\tilde{b}}r_{\tilde{c}}r_{\tilde{d}}\Bigg)\nu^4.
\end{aligned}$$

To simplify programming we linearise $\tilde{\mathfrak{D}}$, to obtain $\mathfrak{D}$, by introducing a global upper bound r_* on r_σ, $\sigma \in \mathcal{X}$. Let σ, ρ, $d \in \mathcal{X}$ then quadratic terms $r_\sigma r_\rho$ will be bounded by $\frac{1}{2}r_*r_\sigma + \frac{1}{2}r_*r_\rho = \frac{1}{2}r_*(r_\sigma + r_\rho)$. For $r_\sigma r_\rho r_d$ we use the bound $\frac{1}{3}r_*^2(r_\sigma + r_\rho + r_d)$. We can do this linearisation because we assumed that the numerically found root is near the exact solution. In the computational step we need to choose a value for r_*. This value must not be too large, nor too small. Choosing r_* too small results in incapability to find an r such that $p(r) < 0$ for $0 \le r_\sigma < r_*$, $\sigma \in \mathcal{X}$, because such an r is nonexistent.

After linearising we obtain $\mathfrak{D}$, to be used in the bound of the tail part $Z^2(r)$:

$$\begin{aligned}
\mathfrak{D}_b &:= r_{\tilde{b}}\nu^2,\\
\mathfrak{D}_{bc} &:= (|c_0|r_{\tilde{b}} + |b_0|r_{\tilde{c}})\nu^2 + \left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{b}} + \frac{1}{2}r_*(r_{\tilde{b}} + r_{\tilde{c}})\right)\nu^3,\\
\mathfrak{D}_{bcd} &:= (|c_0d_0|r_{\tilde{b}} + |b_0d_0|r_{\tilde{c}} + |b_0c_0|r_{\tilde{d}})\nu^2\\
&\quad + \Bigg(|d_0|\left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{b}} + \frac{1}{2}r_*(r_{\tilde{b}} + r_{\tilde{c}})\right)\\
&\qquad + |c_0|\left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{b}} + \frac{1}{2}r_*(r_{\tilde{b}} + r_{\tilde{d}})\right)\\
&\qquad + |b_0|\left(\|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{c}} + \frac{1}{2}r_*(r_{\tilde{c}} + r_{\tilde{d}})\right)\Bigg)\nu^3\\
&\quad + \Bigg(\|\bar{\tilde{b}}\|_{\ell^1_\nu}\|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{b}}\|_{\ell^1_\nu}\|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu}\|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{b}}\\
&\qquad + \frac{1}{2}\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_*(r_{\tilde{c}} + r_{\tilde{d}}) + \frac{1}{2}\|\bar{\tilde{c}}\|_{\ell^1_\nu} r_*(r_{\tilde{b}} + r_{\tilde{d}}) + \frac{1}{2}\|\bar{\tilde{d}}\|_{\ell^1_\nu} r_*(r_{\tilde{b}} + r_{\tilde{c}})\\
&\qquad + \frac{1}{3}r_*^2(r_{\tilde{b}} + r_{\tilde{c}} + r_{\tilde{d}})\Bigg)\nu^4.
\end{aligned}$$

For the remaining, finite part of the bound $Z^2(r)$ (the bound on the part that gets multiplied by $\bar{A}$), we introduce $\mathfrak{C}$. Note that $\mathfrak{C}$ is an N-dimensional vector. We define

$\mathfrak{C}$ element-wise below. Similar to the ODE and DAE example, where we defined C and $C_{\tilde{a}}$, $C_{\tilde{b}}$, respectively, we first define

$$\xi_n := \begin{cases} \nu^{-n} & n \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

As for $\mathfrak{D}$, we also linearise to obtain $\mathfrak{C}$. Using (4.11) and the linearisation using r_* mentioned above, we define, element-wise,

$$\begin{aligned}
(\mathfrak{C}_b)_n &:= r_{\tilde{b}}\xi_{n-2}, \\
(\mathfrak{C}_{bc})_n &:= (|c_0|r_{\tilde{b}} + |b_0|r_{\tilde{c}})\xi_{n-2} + \left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{b}} + \frac{1}{2} r_*(r_{\tilde{b}} + r_{\tilde{c}})\right)\xi_{n-3}, \\
(\mathfrak{C}_{bcd})_n &:= (|c_0 d_0|r_{\tilde{b}} + |b_0 d_0|r_{\tilde{c}} + |b_0 c_0|r_{\tilde{d}})\,\xi_{n-2} \\
&\quad + \Bigg(|d_0|\left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{b}} + \frac{1}{2} r_*(r_{\tilde{b}} + r_{\tilde{c}})\right) \\
&\qquad + |c_0|\left(\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{b}} + \frac{1}{2} r_*(r_{\tilde{b}} + r_{\tilde{d}})\right) \\
&\qquad + |b_0|\left(\|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{c}} + \frac{1}{2} r_*(r_{\tilde{c}} + r_{\tilde{d}})\right)\Bigg)\xi_{n-3} \\
&\quad + \Bigg(\|\bar{\tilde{b}}\|_{\ell^1_\nu}\|\bar{\tilde{c}}\|_{\ell^1_\nu} r_{\tilde{d}} + \|\bar{\tilde{b}}\|_{\ell^1_\nu}\|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{c}} + \|\bar{\tilde{c}}\|_{\ell^1_\nu}\|\bar{\tilde{d}}\|_{\ell^1_\nu} r_{\tilde{b}} \\
&\qquad + \frac{1}{2}\|\bar{\tilde{b}}\|_{\ell^1_\nu} r_*(r_{\tilde{c}} + r_{\tilde{d}}) + \frac{1}{2}\|\bar{\tilde{c}}\|_{\ell^1_\nu} r_*(r_{\tilde{b}} + r_{\tilde{d}}) + \frac{1}{2}\|\bar{\tilde{d}}\|_{\ell^1_\nu} r_*(r_{\tilde{b}} + r_{\tilde{c}}) \\
&\qquad + \frac{1}{3} r_*^2(r_{\tilde{b}} + r_{\tilde{c}} + r_{\tilde{d}})\Bigg)\xi_{n-4}.
\end{aligned}$$

We remark that the formulas for $\mathfrak{C}$ are similar to those of $\mathfrak{D}$. The difference is that ν^2, ν^3, and ν^4 in $\mathfrak{D}$ are replaced by respectively ξ_{n-2}, ξ_{n-3}, and ξ_{n-4} in $\mathfrak{C}$. The following relation between $\mathfrak{C}$ and $\mathfrak{D}$ holds:

$$\mathfrak{D} = \nu^4(\mathfrak{C})_4.$$

Final bound $Z^2(r)$ Using the introduced notation of $\mathfrak{C}$ and $\mathfrak{D}$, we can bound the finite parts as well as the tails. The final bound $Z^2(r)$ becomes, for $\sigma, \rho \in \mathcal{X}$,

$$Z^2_{\rho\sigma} := \sum_{d \in \mathcal{X}} \||\bar{A}_{\rho d}|\bar{C}_{d\sigma}\|_{\ell^1_\nu} + (\hat{A}\hat{Z}^2)_{\rho\sigma}, \tag{4.12}$$

where $\hat{A}$ is defined in Equation (4.10). Furthermore, $(\bar{C}_{d\sigma})_n$ for $0 \leq n \leq N-1$ is defined in Appendix A.2, Table A.2. Formulas for $\hat{Z}^2_{\rho\sigma}$ are given in Table A.3 of Appendix A.3.

Radii polynomial Following the definition of the multi dimensional radii polynomial (Definition 2.11), and extending the alternative representation of (2.11), as presented in (3.23), the radii polynomial is given by

$$p(r) = Y + [Z^0 + Z^1 + Z^2(r) - I]r, \tag{4.13}$$

where I denotes the (6×6) identity matrix and Y, Z^i $i = 0, 1, 2$ are defined in Equations (4.7) to (4.9) and (4.12). We stress that Y is a vector of length 6, and Z^i are (6×6) matrices.

5. Computational results

In Chapter 4, we determined the bounds Y and Z of Theorem 2.12 and constructed the six-dimensional radii polynomial in line with Definition 2.11 and the alternative representation for Z, given in (3.23). The bounds are defined in Equations (4.7) to (4.9) and (4.12) with radii polynomial $p(r)$ given in (4.13). In this chapter, we briefly discuss how one can obtain a numerical solution to system (4.2). Furthermore, we treat the implementation of the computational part of the proof, where we obtain an $r \in (\mathbb{R}^+)^6$ generating balls around the numerical solution in which the exact solution lies. For every power series (4.3), we obtain a corresponding r_σ, $\sigma \in \mathcal{X}$, with the ordered list of variables given by $\mathcal{X} = (\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$.

If we are successful in finding an r such that $p(r) < 0$, element-wise, with a predetermined value for ν, the weight of the ℓ^1_ν-norm (Definition 2.5), then we have validated that an exact, and unique solution to (4.2), for $t \in [0, \nu]$, lies in balls around the numerically found solution. We will demonstrate this computational step in Section 5.2. Furthermore, in Chapter 3, we discussed a procedure to stretch the time interval on which we validate the solution. In this chapter, Section 5.3, we present the results of a similar procedure applied to the more complex differential-algebraic system.

We furthermore remark that, while it was easy to find all r satisfying $p(r) < 0$ in the ODE example of Chapter 3, it already became more challenging to achieve something similar for the DAE example. However, for the DAE example, since the system is two-dimensional, we could still visualise the area where the two-dimensional radii polynomial was negative. For the six-dimensional system that we treat in this chapter, neither are possible. Instead we focus solely on finding the smallest r, satisfying $p(r) < 0$. We elaborate what we mean by 'smallest' in Section 5.2.

5.1. Solving the differential-algebraic system numerically

Making use of the built-in MATLAB ODE solvers that support DAEs, we can solve system (1.5) numerically by using the alternative representation (4.2). One of the solvers that is capable of solving differential-algebraic systems is `ode15i`. We already used this solver to solve the DAE of the second example (Section 3.2). To solve a system using this solver, we need to specify a time interval t_{int}. Furthermore, we need the initial values a_0, b_0, c_0 and the expressions for α_0, β_0, γ_0 given by equation (4.4). We note that, in Subsection 4.1.1, we had two options for the initial value γ_0. Choosing between γ^+ and γ^- is arbitrary. In this chapter we consider only the initial value γ_0^+ to illustrate the computational results and set $\gamma_0 = \gamma_0^+$. After solving the system numerically we obtain estimates for the Taylor coefficients using the built-in function `polyfit`.

Another option to solve system (4.2), is by using the recurrence relations we constructed in Subsection 4.1.1. Observe that we can calculate the Taylor coefficients step by step in the following order: a, b, c, α, γ, β. When viewing the infinite set of equations F, we observe that in every step i we need to determine the coefficient a_i, before we can determine α_i, β_i, γ_i due to dependence on a_i. Similarly, we need to determine γ_i before we can determine β_i.

5.2. Implementation

In the introduction of this chapter, we already mentioned that finding all vectors r satisfying $p(r) < 0$ is not a simple task. For every element p_σ, $\sigma \in \mathcal{X}$, we obtain a polynomial. The goal would then be to find all points $r > 0$ such that every polynomial attains a negative value at r. Instead of finding all r that satisfy $p(r) < 0$ we focus on finding the r that lies 'closest' to the origin and satisfies $p(r) < 0$. We define closest, as closest in the L_1-distance. Similar to the DAE example of Chapter 3 in Equation (3.32), we introduce $\tilde{p}(r)$ as

$$\tilde{p}(r) := \begin{cases} K r_* \sum_{\sigma \in \mathcal{X}} r_\sigma & p(r) < 0, \\ 6 K r_* + \max_{\sigma \in \mathcal{X}} p_\sigma(r) & \text{otherwise,} \end{cases}$$

where r_* is the predetermined upper bound on r_σ, $\sigma \in \mathcal{X}$, as introduced in the paragraph on the bound $Z^2(r)$ in Subsection 4.1.3, and 6 denotes the number of variables in the system. The constant K is a scaling factor to increase numerical accuracy. For our computations, we set $K = 10^6$. The goal of the computational step is then transformed into minimising $\tilde{p}(r)$ for $r > 0$. By minimising, we hope to find the global minimum of $\tilde{p}(r)$, i.e. we hope to find the $r > 0$ closest to the origin in the L_1-distance such that $p(r) < 0$.

Finding global minimiser Finding the global minimiser of $\tilde{p}(r)$ is far from trivial. In an attempt to find the numerical global minimiser we use the `GlobalSearch` algorithm, part of the *Global Optimization Toolbox* of MATLAB in conjunction with the built-in MATLAB function `fmincon`, which minimises a function given constraints. We do note, however, that using the `GlobalSearch` algorithm can drastically increase running time. For the continuation results, discussed in Section 5.3, it is of great importance to find the smallest r in order to be successful. Since one is not always interested in finding the smallest r, but is satisfied with any r we built an option to enable searching globally. The default is not to search for a global minimiser of $\tilde{p}(r)$.

Results[1] We compared run times of searching globally for a minimiser of $\tilde{p}(r)$ to not searching globally. We also compared the resulting vectors r found by the script. As an example, we consider the system 4.2 with the following parameters

$$K_1 = 0.5, \quad K_{\text{eq}} = 0.4, \quad \underline{v} = 0.8, \quad L = 0.2, \tag{5.1}$$

and initial values

$$a_0 = 0.4, \quad b_0 = 0.7, \quad c_0 = 0.3. \tag{5.2}$$

Using the method that uses the built-in MATLAB solver `ode15i` with $t_{\text{int}} = [0, 5]$, explained in Section 5.1, we solve this system numerically. We set $N = 10$ and obtain the approximate Taylor coefficients for this system. We attempt to find an r satisfying $p(r) < 0$. To do so, we set $\nu = 0.1$ and $r_* = 0.1$. Searching globally, and non-globally, for a minimiser, we find respectively

$$r_{\text{glob}} \approx \begin{pmatrix} 5.505 * 10^{-4} \\ 6.926 * 10^{-4} \\ 7.681 * 10^{-4} \\ 1.257 * 10^{-3} \\ 5.178 * 10^{-3} \\ 1.396 * 10^{-3} \end{pmatrix}, \quad r_{\text{loc}} \approx \begin{pmatrix} 7.352 * 10^{-4} \\ 1.102 * 10^{-3} \\ 1.102 * 10^{-3} \\ 1.564 * 10^{-3} \\ 3.053 * 10^{-2} \\ 1.524 * 10^{-3} \end{pmatrix}.$$

We obtain significantly lower values for r when searching globally. However, this does come with a cost. For this specific example the running time doubled.

5.3. Continuation[2]

In the DAE and ODE examples of Chapter 3, we mentioned that to stretch the time interval t_{int} we need to improve the accuracy of the numerical solution. One can obtain a more accurate numerical solution by

- increasing N,
- increasing the computational accuracy of the Taylor coefficients.

An increase in the computational accuracy can be achieved by implementing the multivariate Newton method. However, we discovered that the standard MATLAB solvers do not produce accurate enough results for this Newton method to converge.

To illustrate the concept of continuation, i.e. stretching the time interval t_{int}, we instead use the recurrence relations directly to obtain a numerical solution to system (4.2). We follow a similar procedure as discussed in the continuation paragraph of Subsection 3.1.1. We extend this procedure to our six-dimensional system;

1. Find an accurate numerical solution,

[1]Script used: `script6D.m`
[2]Scripts used: `continuation6D.m`, `continuation6D_fig5_2.m`

2. Choose a ν and validate numerical solution for $t \in [0, \nu]$,
3. Attempt to find the global minimiser $r > 0$ of $\tilde{p}(r)$,
4. Determine six new starting points by using the Taylor series expansion, e.g. $a_0' = \sum_{n=0}^{N} a_n \nu^n \pm r_{\tilde{a}}$,
5. Repeat steps 1-4 for the starting points obtained in step 4.

Observe that the new starting points, obtained in step 4, are intervals and not values. After the first iteration of the above mentioned procedure we are, thus, working with a numerical solution that is given by intervals.

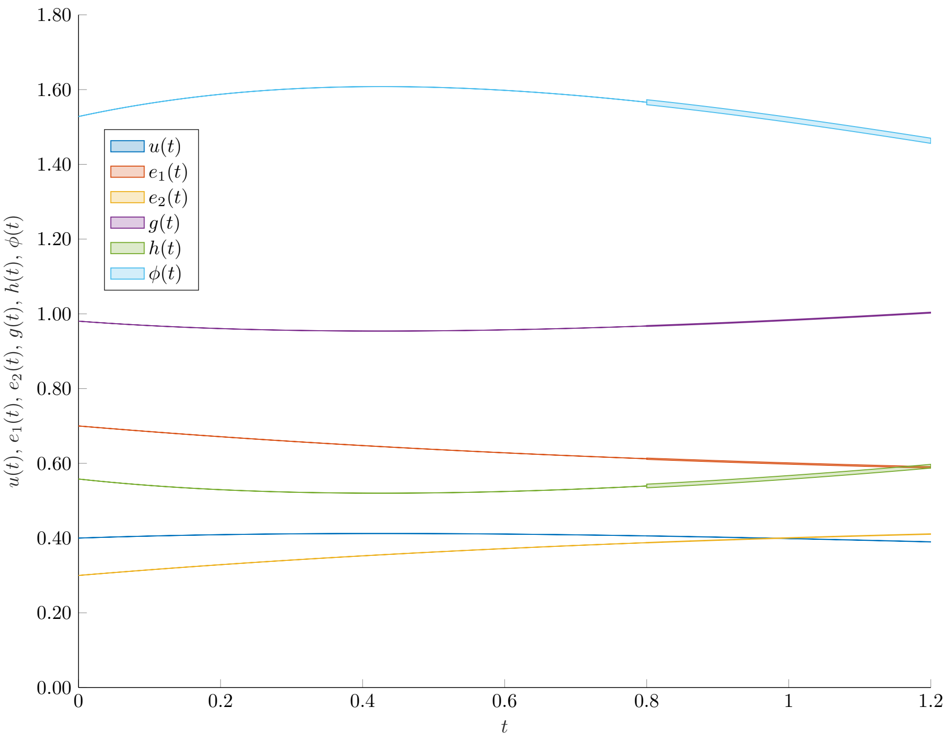


Figure 5.1.: Figure illustrating the area where the exact solution should be, based on the computed numerical guess and the 'smallest' r that we determined.

We choose the same parameters as mentioned in the 'results' paragraph of Section 5.2, given in Equations (5.1) and (5.2). After some experimentation with the code, we set $N = 25$, $\nu = 0.4$, and $r_* = 0.01$. We manage to verify the numerical solution, determined using the recurrence relations explained in Section 5.1, on the interval $t_{\text{int}} = [0, 1.2]$. In other words, we managed to verify the solution for three time steps of size $\nu = 0.4$. After the third iteration, the numerical solution is no longer accurate enough for the current setup to find a positive radii vector r satisfying the component-wise negativity of the radii polynomial.

As in the examples of Chapter 3, we can plot the validated solution using the upper bound (3.17) for all six variables of the system. The validated solution is visualised in Figure 5.1. Furthermore, we visualised the dynamics of the system with parameters (5.1) and initial values (5.2) in Figure 5.2, where we used the built-in MATLAB ODE solver `ode15i` to obtain a numerical solution to the time interval $[0, 7]$. The dashed line in this figure represents the point up to where we managed to validate the solution.

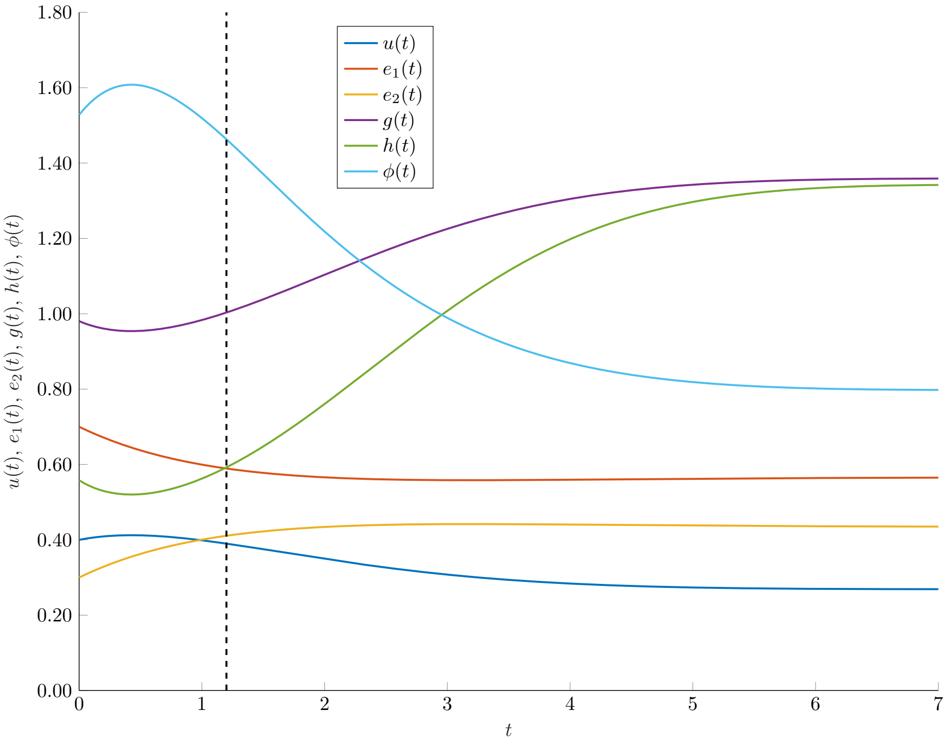


Figure 5.2.: Figure illustrating the the dynamics of the system with parameters (5.1) and initial values (5.2). We managed to validate the solution for $t \in [0, 1.2]$, i.e. for the part from $t = 0$ up to the dashed line at $t = 1.2$.

6. Conclusion

We investigated a method to validate numerical solutions to differential-algebraic systems in systems biology. In particular were we interested in determining whether we could create a ball around a numerical solution in which an exact, and unique, solution to a differential-algebraic system lies. To demonstrate this we considered a simplified toy model of the qORAC framework described in [11]. Based on the existing theory in [5], we were able to construct a theorem, that works in a differential-algebraic setting, in Section 2.6. This theorem introduces conditions that guarantee the existence of an exact, and unique, solution within balls, with certain radii, around the numerical solution.

Accomplishments With the developed theory at hand, we first considered two basic examples in Chapter 3 to illustrate the main concepts. In Chapter 4, we transformed the simplified toy model of the qORAC framework, which we introduced in Chapter 1, in an infinite set of equations by using the Taylor series expansion. Due to this transformation, we were able to work in a product space of spaces of Taylor coefficients ℓ_ν^1.

In Chapter 2, we have shown that ℓ_ν^1 is a Banach space and constructed a theorem that specifies conditions guaranteeing the existence of an exact, and unique, solution in a product space consisting of Banach spaces ℓ_ν^1. Having transformed the toy model using the Taylor series expansion enabled us to apply this theorem. In the remainder of Chapter 4, we determined bounds that would work with the theorem of Section 2.6 and finished with the construction of the *radii polynomial*.

The construction of all necessary bounds and its corresponding radii polynomial allowed us to create an implementation in MATLAB. The MATLAB toolbox INTLAB enabled us to construct reliable computations. In an attempt to find the r that lies closest to the origin in the L_1-distance and satisfies the negativity constraint on the radii polynomial, we constructed a function $\tilde{p}(r)$ to be minimised. The resulting r creates the strongest bound on where the exact solution lies.

The validation only holds for the time interval $[0, \nu]$, where ν is a chosen parameter used in the ℓ_ν^1-norm. In the ODE example of Chapter 3, we were able to determine the radius of convergence (an upper bound for ν). However, for more complex systems it is far from trivial to determine this upper bound. Aside from the radius of convergence, the maximum value of ν that works for our implementation depends on many factors. The factors influencing the size of the time interval $[0, \nu]$ are

- the accuracy of the numerical solution,
- the number N of Taylor coefficients calculated,
- the accuracy of the Taylor coefficients,
- the accuracy of the computer used and the accuracy of its computations,

- the tightness of the bounds determined in Chapter 4, and
- the radii of convergence of the used power series.

We attempted to overcome these factors using a continuation method, where one would validate time interval $[0, \nu]$ and determine new initial values. Using these new initial values we attempt to validate the next step $[\nu, 2\nu]$. This continuation method is limited as the error increases with every step. After the first iteration we have validated that the exact solution lies in balls with radii r around the numerical starting solution. We thereby obtain an interval in which the initial values lie for the next iteration. Obviously those intervals increase with every step, and thereby our chances to accomplish validation decrease.

In conclusion, we demonstrated that the theory of [5], presented for ODE systems, can be extended to differential-algebraic systems consisting of equations of polynomial form. To achieve this, we constructed a general theory for a product space of Taylor coefficients, where the dimension of this product space equals the number of variables in the differential-algebraic system. The constructed theorem allows us to obtain an explicit range in which an exact solution of a differential-algebraic system lies based on a numerical solution of the system. We illustrated the power of the theorem through a toy model of the qORAC framework obtained from the systems biology field and, in the process, introduced all necessities to apply the validation technique to arbitrary DAEs of polynomial form.

Future research For future research, it is of interest to further investigate the validation of larger time intervals. In this thesis, the maximal time interval for which we can validate a solution is limited by many factors, mentioned in the previous paragraph. To remove, or decrease, this restriction one can for instance consider to use a different series representation to transform the dynamic problem of interest into a root-finding problem. Series to consider could be the Lagrange, Bernstein, and Newton polynomials. One can also attempt to create tighter bounds Y, Z^0, Z^1, and Z^2. However, we do note that tighter bounds are more complex and result in an increase in the complexity of the computational part (programming the bounds gets harder) as well as an increase in complexity of the analytical derivation of the bounds.

Automating the analytical derivation of the bounds of Chapter 4 allows us to extend the presented validation method to arbitrary differential-algebraic systems that can be written in polynomial form. We emphasise that the symbolic toolbox of MATLAB will prove useful in converting such a differential-algebraic system into a differential-algebraic system in polynomial form. Furthermore, we believe that we worked out all necessary (mathematical) details to enable one to automate the analytical derivation of the bounds of Chapter 4. The challenge of this automation lies largely in the programming aspect. For the computational step of the proof, we note that the MATLAB functions that we constructed can either be reused or can be easily adapted to work for arbitrary systems.

A. Bounds

We present the bounds $\hat{Z}^1$, $\bar{C}$, and $\hat{Z}^2$.

A.1. $\hat{Z}^1$

	$\tilde{a}$	$\tilde{b}$	$\tilde{c}$
$\tilde{a}$	$\lvert \underline{v} - LK_{\text{eq}} \rvert \mathfrak{B}_{b\alpha} + 2K_{\text{eq}}\mathfrak{B}_{ab\alpha} + (K_1 + \underline{v})\mathfrak{B}_{c\alpha} + 2\mathfrak{B}_{ac\alpha}$	$\underline{v}L\mathfrak{B}_{\alpha} + \lvert \underline{v} - LK_{\text{eq}} \rvert \mathfrak{B}_{a\alpha} + K_{\text{eq}}\mathfrak{B}_{aa\alpha}$	$(K_1 + \underline{v})\mathfrak{B}_{a\alpha} + \mathfrak{B}_{aa\alpha}$
$\tilde{b}$	$K_1\mathfrak{B}_{\beta} + 2\mathfrak{B}_{a\beta} + \mathfrak{B}_{\beta\gamma}$	ν	0
$\tilde{c}$	$K_1\mathfrak{B}_{\beta} + 2\mathfrak{B}_{a\beta} + \mathfrak{B}_{\beta\gamma}$	0	ν
$\tilde{\alpha}$	$(K_1 + \underline{v} + L)\left(\frac{\mathfrak{B}_{\alpha}}{\nu} - \alpha_0\right) + 2\left(\frac{\mathfrak{B}_{a\alpha}}{\nu} - a_0\alpha_0\right)$	0	0
$\tilde{\beta}$	$\lvert K_1 - LK_{\text{eq}} \rvert \left(\frac{\mathfrak{B}_{\beta}}{\nu} - \beta_0\right) + 2\left(\frac{\mathfrak{B}_{\beta\gamma}}{\nu} - \beta_0\gamma_0\right)$ $+2\lvert 1 - K_{\text{eq}} \rvert \left(\frac{\mathfrak{B}_{a\beta}}{\nu} - a_0\beta_0\right)$	0	0
$\tilde{\gamma}$	$2(1 + K_{\text{eq}})\left(\frac{\mathfrak{B}_{a\gamma}}{\nu} - a_0\gamma_0\right) + 2\lvert K_1K_{\text{eq}} - LK_{\text{eq}}^2 \rvert \left(\frac{\mathfrak{B}_{a}}{\nu} - a_0\right)$ $+2LK_{\text{eq}}\left(\frac{\mathfrak{B}_{\gamma}}{\nu} - \gamma_0\right)$	0	0

	$\tilde{\alpha}$	$\tilde{\beta}$	$\tilde{\gamma}$
$\tilde{a}$	$\underline{v}L\mathfrak{B}_{b} + \lvert \underline{v} - LK_{\text{eq}} \rvert \mathfrak{B}_{ab} + K_{\text{eq}}\mathfrak{B}_{aab} + (K_1 + \underline{v})\mathfrak{B}_{ac} + \mathfrak{B}_{aac}$	0	0
$\tilde{b}$	0	$K_1\mathfrak{B}_{a} + \mathfrak{B}_{aa} + \mathfrak{B}_{a\gamma}$	$\mathfrak{B}_{a\beta}$
$\tilde{c}$	0	$K_1\mathfrak{B}_{a} + \mathfrak{B}_{aa} + \mathfrak{B}_{a\gamma}$	$\mathfrak{B}_{a\beta}$
$\tilde{\alpha}$	$(K_1 + \underline{v} + L)\left(\frac{\mathfrak{B}_{a}}{\nu} - a_0\right) + \frac{\mathfrak{B}_{aa}}{\nu} - a_0^2$	0	0
$\tilde{\beta}$	0	$\lvert K_1 - LK_{\text{eq}} \rvert \left(\frac{\mathfrak{B}_{a}}{\nu} - a_0\right) + L\left(\frac{\mathfrak{B}_{\gamma}}{\nu} - \gamma_0\right)$ $+2\left(\frac{\mathfrak{B}_{a\gamma}}{\nu} - a_0\gamma_0\right) + \lvert 1 - K_{\text{eq}} \rvert \left(\frac{\mathfrak{B}_{aa}}{\nu} - a_0^2\right)$	$L\left(\frac{\mathfrak{B}_{\beta}}{\nu} - \beta_0\right) + 2\left(\frac{\mathfrak{B}_{a\beta}}{\nu} - a_0\beta_0\right)$
$\tilde{\gamma}$	0	0	$(1 + K_{\text{eq}})\left(\frac{\mathfrak{B}_{aa}}{\nu} - a_0^2\right) + 2L\left(\frac{\mathfrak{B}_{\gamma}}{\nu} - \gamma_0\right)$ $+2LK_{\text{eq}}\left(\frac{\mathfrak{B}_{a}}{\nu} - a_0\right)$

Table A.1.: Table containing bounds $\hat{Z}^1$. One should read this table as follows: the upper-left bound represents $\hat{Z}^1_{\tilde{a}\tilde{a}}$, the one next to it represents $\hat{Z}^1_{\tilde{a}\tilde{b}}$, the one underneath it represents $\hat{Z}^1_{\tilde{b}\tilde{a}}$, etc.

A.2. $\bar{C}$

	$\tilde{a}$	$\tilde{b}$	$\tilde{c}$
$\tilde{a}$	$\lvert\underline{v} - LK_{\text{eq}}\rvert\,(\mathfrak{C}_{b\alpha})_n + 2K_{\text{eq}}\,(\mathfrak{C}_{ab\alpha})_n + (K_1 + \underline{v})\,(\mathfrak{C}_{c\alpha})_n + 2\,(\mathfrak{C}_{ac\alpha})_n$	$\underline{v}L\,(\mathfrak{C}_{\alpha})_n + \lvert\underline{v} - LK_{\text{eq}}\rvert\,(\mathfrak{C}_{a\alpha})_n + K_{\text{eq}}\,(\mathfrak{C}_{aa\alpha})_n$	$(K_1 + \underline{v})\,(\mathfrak{C}_{a\alpha})_n + (\mathfrak{C}_{aa\alpha})_n$
$\tilde{b}$	$K_1\,(\mathfrak{C}_{\beta})_n + 2\,(\mathfrak{C}_{a\beta})_n + (\mathfrak{C}_{\beta\gamma})_n$	0	0
$\tilde{c}$	$K_1\,(\mathfrak{C}_{\beta})_n + 2\,(\mathfrak{C}_{a\beta})_n + (\mathfrak{C}_{\beta\gamma})_n$	0	0
$\tilde{\alpha}$	$(K_1 + \underline{v} + L)\,(\mathfrak{C}_{\alpha})_{n+1} + 2\,(\mathfrak{C}_{a\alpha})_{n+1}$	0	0
$\tilde{\beta}$	$\lvert K_1 - LK_{\text{eq}}\rvert\,(\mathfrak{C}_{\beta})_{n+1} + 2\,(\mathfrak{C}_{\beta\gamma})_{n+1} + 2\,\lvert 1 - K_{\text{eq}}\rvert\,(\mathfrak{C}_{a\beta})_{n+1}$	0	0
$\tilde{\gamma}$	$2(1 + K_{\text{eq}})\,(\mathfrak{C}_{a\gamma})_{n+1} + 2\,\lvert K_1K_{\text{eq}} - LK_{\text{eq}}^2\rvert\,(\mathfrak{C}_{a})_{n+1} + 2LK_{\text{eq}}\,(\mathfrak{C}_{\gamma})_{n+1}$	0	0

	$\tilde{\alpha}$	$\tilde{\beta}$	$\tilde{\gamma}$
$\tilde{a}$	$\underline{v}L\,(\mathfrak{C}_{b})_n + \lvert\underline{v} - LK_{\text{eq}}\rvert\,(\mathfrak{C}_{ab})_n + K_{\text{eq}}\,(\mathfrak{C}_{aab})_n + (K_1 + \underline{v})\,(\mathfrak{C}_{ac})_n$ $+ (\mathfrak{C}_{aac})_n$	0	0
$\tilde{b}$	0	$K_1\,(\mathfrak{C}_{a})_n + (\mathfrak{C}_{aa})_n + (\mathfrak{C}_{a\gamma})_n$	$(\mathfrak{C}_{a\beta})_n$
$\tilde{c}$	0	$K_1\,(\mathfrak{C}_{a})_n + (\mathfrak{C}_{aa})_n + (\mathfrak{C}_{a\gamma})_n$	$(\mathfrak{C}_{a\beta})_n$
$\tilde{\alpha}$	$(K_1 + \underline{v} + L)\,(\mathfrak{C}_{a})_{n+1} + (\mathfrak{C}_{aa})_{n+1}$	0	0
$\tilde{\beta}$	0	$\lvert K_1 - LK_{\text{eq}}\rvert\,(\mathfrak{C}_{a})_{n+1} + L\,(\mathfrak{C}_{\gamma})_{n+1} + 2\,(\mathfrak{C}_{a\gamma})_{n+1}$ $+ \lvert 1 - K_{\text{eq}}\rvert\,(\mathfrak{C}_{aa})_{n+1}$	$L\,(\mathfrak{C}_{\beta})_{n+1} + 2\,(\mathfrak{C}_{a\beta})_{n+1}$
$\tilde{\gamma}$	0	0	$(1 + K_{\text{eq}})\,(\mathfrak{C}_{aa})_{n+1} + 2L\,(\mathfrak{C}_{\gamma})_{n+1}$ $+2LK_{\text{eq}}\,(\mathfrak{C}_{a})_{n+1}$

Table A.2.: Table containing bounds $\bar{C}$. One should read this table as follows: the upper-left bound represents $(\bar{C}_{\tilde{a}\tilde{a}})_n$, the one next to it represents $(\bar{C}_{\tilde{a}\tilde{b}})_n$, the one underneath it represents $(\bar{C}_{\tilde{b}\tilde{a}})_n$, etc.

A.3. $\hat{Z}^2$

	$\tilde{a}$	$\tilde{b}$	$\tilde{c}$
$\tilde{a}$	$\lvert\underline{v} - LK_{\text{eq}}\rvert\,\mathfrak{D}_{b\alpha} + 2K_{\text{eq}}\mathfrak{D}_{ab\alpha} + (K_1 + \underline{v})\mathfrak{D}_{c\alpha} + 2\mathfrak{D}_{ac\alpha}$	$\underline{v}L\mathfrak{D}_{\alpha} + \lvert\underline{v} - LK_{\text{eq}}\rvert\,\mathfrak{D}_{a\alpha} + K_{\text{eq}}\mathfrak{D}_{aa\alpha}$	$(K_1 + \underline{v})\mathfrak{D}_{a\alpha} + \mathfrak{D}_{aa\alpha}$
$\tilde{b}$	$K_1\mathfrak{D}_{\beta} + 2\mathfrak{D}_{a\beta} + \mathfrak{D}_{\beta\gamma}$	0	0
$\tilde{c}$	$K_1\mathfrak{D}_{\beta} + 2\mathfrak{D}_{a\beta} + \mathfrak{D}_{\beta\gamma}$	0	0
$\tilde{\alpha}$	$(K_1 + \underline{v} + L)\frac{\mathfrak{D}_{\alpha}}{\nu} + 2\frac{\mathfrak{D}_{a\alpha}}{\nu}$	0	0
$\tilde{\beta}$	$\lvert K_1 - LK_{\text{eq}}\rvert\,\frac{\mathfrak{D}_{\beta}}{\nu} + 2\frac{\mathfrak{D}_{\beta\gamma}}{\nu} + 2\,\lvert 1 - K_{\text{eq}}\rvert\,\frac{\mathfrak{D}_{a\beta}}{\nu}$	0	0
$\tilde{\gamma}$	$2(1 + K_{\text{eq}})\frac{\mathfrak{D}_{a\gamma}}{\nu} + 2\,\lvert K_1K_{\text{eq}} - LK_{\text{eq}}^2\rvert\,\frac{\mathfrak{D}_{a}}{\nu} + 2LK_{\text{eq}}\frac{\mathfrak{D}_{\gamma}}{\nu}$	0	0

	$\tilde{\alpha}$	$\tilde{\beta}$	$\tilde{\gamma}$
$\tilde{a}$	$\underline{v}L\mathfrak{D}_{b} + \lvert\underline{v} - LK_{\text{eq}}\rvert\,\mathfrak{D}_{ab} + K_{\text{eq}}\mathfrak{D}_{aab} + (K_1 + \underline{v})\mathfrak{D}_{ac}$ $+\mathfrak{D}_{aac}$	0	0
$\tilde{b}$	0	$K_1\mathfrak{D}_{a} + \mathfrak{D}_{aa} + \mathfrak{D}_{a\gamma}$	$\mathfrak{D}_{a\beta}$
$\tilde{c}$	0	$K_1\mathfrak{D}_{a} + \mathfrak{D}_{aa} + \mathfrak{D}_{a\gamma}$	$\mathfrak{D}_{a\beta}$
$\tilde{\alpha}$	$(K_1 + \underline{v} + L)\frac{\mathfrak{D}_{a}}{\nu} + \frac{\mathfrak{D}_{aa}}{\nu}$	0	0
$\tilde{\beta}$	0	$\lvert K_1 - LK_{\text{eq}}\rvert\,\frac{\mathfrak{D}_{a}}{\nu} + L\frac{\mathfrak{D}_{\gamma}}{\nu} + 2\frac{\mathfrak{D}_{a\gamma}}{\nu}$ $+\,\lvert 1 - K_{\text{eq}}\rvert\,\frac{\mathfrak{D}_{aa}}{\nu}$	$L\frac{\mathfrak{D}_{\beta}}{\nu} + 2\frac{\mathfrak{D}_{a\beta}}{\nu}$
$\tilde{\gamma}$	0	0	$(1 + K_{\text{eq}})\frac{\mathfrak{D}_{aa}}{\nu} + 2L\frac{\mathfrak{D}_{\gamma}}{\nu}$ $+2LK_{\text{eq}}\frac{\mathfrak{D}_{a}}{\nu}$

Table A.3.: Table containing bounds $\hat{Z}^2$. One should read this table as follows: the upper-left bound represents $\hat{Z}^2_{\tilde{a}\tilde{a}}$, the one next to it represents $\hat{Z}^2_{\tilde{a}\tilde{b}}$, the one underneath it represents $\hat{Z}^2_{\tilde{b}\tilde{a}}$, etc.

B. MATLAB

We present the created MATLAB functions and scripts. Toolboxes needed to run the scripts and functions:

- INTLAB (change `dir` in `start_intlab.m`)
- Optimization Toolbox
- Symbolic Math Toolbox
- Global Optimization Toolbox (only necessary when using GlobalSearch)

Short description of the functions and scripts:

General functions and scripts

`bmmulti.m`	Block matrix multiplication
`convo.m`	Calculates the convolution
`matrixnorml1nu.m`	Calculates upper bound of operator norm (Proposition 2.10)
`norml1nu.m`	Calculates ℓ^1_ν-norm
`p_tilde.m`	Used to find an r such that $p(r) < 0$
`start_intlab.m`	Starts INTLAB, change `dir` to location of INTLAB
`test_rp.m`	Tests rigorously the validity of the returned r

Functions and scripts one-dimensional example

`continuation1D.m`	Runs the continuation for the ODE example
`ex1D.m`	Computational step for the ODE example
`script1D.m`	Solve the ODE and validates for time period $[0, \nu]$

Functions and scripts two-dimensional example

`continuation2D.m`	Runs the continuation for the DAE example
`ex2D.m`	Computational step for the DAE example
`images_ex2D.m`	Plot Figure 3.3
`script2D.m`	Solve the DAE and validates for time period $[0, \nu]$

Functions and scripts six-dimensional system

`continuation6D.m`	Runs the continuation for the qORAC toy model
`continuation6D_fig5_2.m`	Plots Figure 5.2
`ex6D.m`	Computational step for the qORAC toy model
`script6D.m`	Solve the qORAC toy model and validates for time period $[0, \nu]$

Bibliography

[1] Ainsworth, S., *Steady-State Enzyme Kinetics*, Palgrave; 43-73, 1977.

[2] Almezel, S., Ansari, Q.H., Khamsi, M.A., *Topics in Fixed Point Theory*, Springer International Publishing; 33-39, 2014.

[3] Ambrosetti, A., Prodi, G., *A Primer of Nonlinear Analysis*, Cambridge University Press; 9-14, 1995.

[4] Basan, M., Hui, S., Okano, H., Zhang, Z., Shen, Y., Williamson, J.R., Hwa, T., *Overflow metabolism in Escherichia coli results from efficient proteome allocation*, Nature 528; 99104, 2015.

[5] Berg, G.J.B. van den, *Introduction to Rigorous Numerics in Dynamics: General Functional Analytic Setup and an Example that Forces Chaos*, Preprint, 2017.

[6] Berg, J.M., Tymoczko, J.L., Stryer, L., *Biochemistry. 5th edition. Section 8.3, Enzymes Accelerate Reactions by Facilitating the Formation of the Transition*, W H Freeman, 2002.

[7] Bosdriesz, E., Molenaar, D., Teusink, B., Bruggeman, F. J., *How fast-growing bacteria robustly tune their ribosome concentration to approximate growth-rate maximization*, The Febs Journal, 282(10); 20292044, 2015.

[8] Cooper, G.M., *The Cell: A Molecular Approach. 2nd edition. The Central Role of Enzymes as Biological Catalysts*, Sunderland (MA): Sinauer Associates, 2000.

[9] Lohr, D., Venkov, P., and Zlatanaoca, J., *Transcriptional regulation in the yeast GAL gene family: a complex genetic network*, faseb, 9(9); 777-787, 1995.

[10] Molenaar, D., Berlo, R. van, Ridder, D. de, Teusink, B., *Shifts in growth strategies reflect tradeoffs in cellular economics*, Molecular Systems Biology **5**, 2009.

[11] Planqué, R., Hulshof, J., Teusink, B., Hendriks, J.C., Bruggeman, F.J., *Cells can maintain optimal metabolism using simple regulation*, Unpublished, 2017.

[12] Ray, J.C.J., Wickersheim, M.L., Jalihal, A.P., Adeshina, Y.O., Cooper, T.F., Balzsi, G., *Cellular Growth Arrest and Persistence from Enzyme Saturation*, PLoS Computational Biology, 12(3), 2016.

[13] Rump, S.M., *Developments in Reliable Computing*, Kluwer Academic Publishers; 77-104, 1999.

[14] Scott, M., Klumpp, S., Mateescu, E.M., Zhang, Z., Hwa, T., *Emergence of robust growth laws from optimal regulation of ribosome synthesis*, Molecular Systems Biology 10, 2014.

[15] Scott, M., Gunderson, C.W., Mateescu, E.M., Hwa, T., *Interdependence of cell growth and gene expression: origins and consequences*, Science 330;1099-1102, 2010.

[16] Steen, L.A., Seebach, J.A., *Counterexamples in Topology*, Courier Corporation; 56, 1995.