TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DIPLOMARBEIT

Numerical treatment of models from applications using `bvpsuite`

Ausgeführt am Institut für
Analysis und Scientific Computing
der Technischen Universität Wien

unter der Anleitung von
Ao. Univ. Prof. Dr. Ewa B. Weinmüller

durch

Anna Feichtinger, BSc.
Wien

Wien, am 15. Oktober 2014

Acknowledgments

I would like to express my very great appreciation to Ao. Univ. Prof. Dr. Ewa Weinmüller for her patient guidance, enthusiastic encouragement and useful criticism of this work.

My thanks and appreciations also go to Prof. RNDr. Irena Rachåunková, DrSc. and Prof. RNDr. Svatoslav Staněk, CSc. from the Department of Mathematics of the Palacky University in Olomouc, Czech Republic, for the opportunity to contribute to their interesting work.

My special thanks go to Univ.Prof. Dipl.-Ing. Dr.techn. Anton Friedl, Ass.Prof. Dipl.-Ing. Dr.techn. Michael Harasek, and Dipl.-Ing. Dr.techn. Aleksander Makaruk from the Institute for Chemical Engineering of the Vienna University of Technology, for sharing their knowledge on gas permeation with me.

Finally, I would like to thank my Family and Friends for their understanding and support in completing this project.

Abstract

In this work we deal with the method of polynomial collocation and use it to solve boundary value problems in ordinary differential equations. Collocation was implemented in the open domain MATLAB code `bvpsuite` which has been developed at the Institute for Analysis and Scientific Computing, Vienna University of Technology. This implementation includes an error estimate and an adaptive mesh selection which enhances the efficiency of the code will be described in detail later.

In this master thesis we present two classes of applications simulated using `bvpsuite`. The first application is a project in cooperation with the Department of Mathematics of the Palacky University in Olomouc, Czech Republic. It is concerned with the existence of solutions of nonlinear singular second order ordinary differential equation of the form

$$u''(t) = \frac{a}{t}u'(t) + \lambda f(t, u(t), u'(t)), \quad t \in (0, T)\,,$$

subject to periodic boundary conditions

$$u(0) = u(T),\ u'(0) = u'(T).$$

We illustrate this theory by means of numerical simulations.

The second application is a simulation of a model for gas permeation, in cooperation with the Institute for Chemical Engineering of the Vienna University of Technology. It shows how using `bvpsuite` allows to handle difficult models which cannot be treated effectively using standard techniques.

Contents

1 Introduction

1.1 Definitions

An *ordinary differential equation* (ODE) is an equation in which the values of the solution z are linked to the values of its derivatives $z^{(k)}$, $1 \leq k \leq n$. Many models from natural sciences and engineering can be expressed by systems of differential equations and therefore, ODEs are important in many scientific disciplines, like physics, mechanics or chemistry.

The famous Newton's second law has the form $z''(t) = f(t)/m$, where f is the total force acting on an object, m is the mass of the object and z'' is the acceleration. This equation is said to be of *second order* because the second derivative is the highest derivative appearing in the equation.

Definition 1.1.1 *A differential equation*

$$F(t, z(t), z'(t), \dots, z^{(n)}(t)) = 0, \quad t \in [a, b], \tag{1.1}$$

is called implicit. A differential equation is called explicit if it takes the form

$$z^{(n)}(t) = f(t, z(t), z'(t), \dots, z^{(n-1)}(t)). \tag{1.2}$$

The order of an differential equation is the highest derivative appearing in the equation. A function $z(t)$ is called a solution of (1.1) on the interval $I \subset [a, b]$, if z is n times continuously differentiable and satisfies the differential equation on I.

Usually, solution of the ODE systems (1.1) or (1.2) are not unique. Therefore, we have to prescribe additional conditions the unique solution has to satisfy. This is especially important when we attempt to solve the problem numerically. In this case the solution of the problem has to be unique, or at least locally unique. In this context, we have two typical problem settings, *initial value problems* (IVPs) and *boundary value problems* (BVPs).

Definition 1.1.2 *A differential equation subject to initial conditions is called an initial value problem and takes the form*

$$F(t, z(t), z'(t), \dots, z^{(n)}(t)) = 0, \quad z(t_0) = z_0,\, z'(t_0) = z_1,\, \dots,\, z^{(n-1)}(t_0) = z_{n-1}, \;\; t \in [a, b].$$

The values $z_0, \dots, z_{n-1}$ are called initial values. If the additional requirements are not posed at the same point t_0, but at different points, we obtain a boundary value problem and the corresponding values are called boundary values.

Definition 1.1.3 *An ordinary differential equation*

$$z'(t) = \frac{\lambda}{t^{\alpha}} z(t) + g(t, z(t)), \quad t \in (0, 1],$$

where λ is a given constant and g a given function, is called singular. If $0 < \alpha < 1$, we call the singularity week, we refer to a singularity of the first kind if $\alpha = 1$, while for $\alpha > 1$ the singularity is of second kind or essential.

1.2 Aim and structure of this master thesis

Systems of differential equations in form of IVPs and BVPs often arise in models from natural sciences and engineering. Most of the mathematical models describing the applications cannot be solved exactly and therefore, an approximation to the solution derived from a suitable numerical method is the only option. There is a variety of solvers and controlling mechanisms to choose from when designing a code for the numerical solutions of ODE systems. Such open domain codes usually not only provide the approximate solution but also an estimate of its error. Moreover, they attempt to provide the numerical solution with as small computational cost as possible. In this work, we shall apply the open domain MATLAB code `bvpsuite` to solve two problems from mechanics and chemistry. This code is based on polynomial collocation and can cope with singular implicit BVPs of arbitrary order.

This master thesis is organized as follows: In Section 2, we recapitulate the most important analytical properties of regular und singular ODEs. In Section 3 the numerical algorithm used in `bvpsuite` as a basic solver, the error estimate strategy and the mesh selection are discussed. Section 4 contains the numerical simulation of two applications. The first example is a singular ODE, while the second application is a regular ODE modelling the gas separation by permeation through a membrane.

The aim of this master thesis is, first of all to give an overview of the analytical properties of the ODE systems in their regular und singular form. Moreover, we shall discuss the main principles of the respective software design and show how they were implemented in `bvpsuite`. Finally, we use this code to numerically simulate two applications. The first model exhibits a singularity of the first kind and shows that the solver can easily and efficiently cope with this type of difficulty. This model has been simulated within an international cooperation with the Department of Mathematics, Palacky University, Olomouc, Czech Republic. The second model describes a chemical process of gas separation. This project was carried out together with Institute for Chemical Engineering, Vienna University of Technology, Vienna, Austria. Especially, numerical tests of the second problem show how modern, dependable software can help solving involved problems relevant for industrial applications.

2 Analytical results for initial value problems

In this chapter, see [16], we discuss the existence and uniqueness of solutions of the first order initial value problems (IVPs) for ODEs of the form

$$z'(t) = f(t, z(t)), \quad z(t_0) = z_0, \tag{2.1}$$

where $f : G \to \mathbb{R}^n$ is continuous on the open set $G \subseteq \mathbb{R}^{n+1}$ and $(t_0, z_0) \in G$.

2.1 Existence and uniqueness of solutions of regular IVPs

We first state basic analytical results for the regular problems without singularities. Before we present the theorems, we define the Lipschitz continuity of the right-hand side f.

Definition 2.1.1 *The function $f : G \to \mathbb{R}^n$ is Lipschitz continuous in z (with Lipschitz constant L) if*

$$\|f(t, z_1) - f(t, z_2)\| \leq L \|z_1 - z_2\| \text{ for } (t, z_1), (t, z_2) \in G.$$

f is called locally Lipschitz continuous if for every $(t^*, z^*) \in G$ there exists a neighborhood U of (t^*, z^*), such that $f\,|_U$ is Lipschitz continuous.

Theorem 2.1.2 (Picard-Lindelöf) *Assume that $f : G \to \mathbb{R}^n$ is continuous and locally Lipschitz continuous in z on the open set $G \subseteq \mathbb{R}^{n+1}$ and $(t_0, z_0) \in G$. Then, for a $\delta > 0$, there exists a unique function $z(t) \in C^1(J_\delta, \mathbb{R}^n)$ with $J_\delta := [t_0 - \delta, t_0 + \delta]$, so that $(t, z(t)) \in G$ for $t \in J_\delta$ and $z(t)$ is a solution of (2.1) on J_δ.*

If the right-hand side of the differential equation is continuous but not Lipschitz continuous, the following theorem shows the existence of at least one solution of the differential equation.

Theorem 2.1.3 (Peano) *Suppose $G \subseteq \mathbb{R}^{n+1}$ is open and $f : G \to \mathbb{R}^n$ is continuous. Then there exists $\delta > 0$ and a function $z(t) \in C^1(J_\delta, \mathbb{R}^n)$ with $J_\delta := [t_0, -\delta,\, t_0 + \delta]$, so that $(t, z(t)) \in G$ for $t \in J_\delta$ and $z(t)$ is a solution of (2.1) in J_δ.*

Note that the solution in this last theorem is not necessarily unique and that J_δ is the same interval as in the theorem of Picard-Lindelöf.

2.2 Maximum domain of solution, and global uniqueness

Note, that the above theorems only guarantee that a solution exists only on a short interval $[t_0 - \delta,\, t_0 + \delta]$. This solutions can be extended to an greater interval.

2.2.1 Extension of a solution

Let $G \subseteq \mathbb{R}^{n+1}$ be open and $f : G \to \mathbb{R}^n$ Lipschitz continuous in z. For $(t_0, z_0) \in G$, Theorem 2.1.2 guarantees a solution $z_0(t)$ in a potentially small interval $J_0 = [t_0 - \delta_0,\, t_0 + \delta_0]$. Let $t_1 := t_0 + \delta_0$ and $z_1 := z_0(t_1)$. Due to Theorem 2.1.2, $(t_1, z_1) \in G$ and the initial value problem (2.1) with the boundary condition $z(t_1) = z_1$ has a unique solution $z_1(t)$ on the interval $J_1 := [t_1 - \delta_1,\, t_1 + \delta_1]$ with $\delta_1 > 0$. Hence, both $z_0(t)$ and $z_1(t)$ are solutions of the same differential equation and thus, they coincide on $t \in J_0 \cap J_1$. Now we can define a solution on the interval $[t_0,\, t_1 + \delta_1]$:

$$z_+(t) := \begin{cases} z_0(t), & t \in [t_0, t_1], \\ z_1(t), & t \in [t_1,\, t_1 + \delta_1]. \end{cases}$$

The solution $z_+(t)$ is called the *extension to the right*. It is possible to define a *extension to the left* in an analogous way.

In principle, we could repeat this process to further enlarge the solvability interval. Unfortunately, this procedure does not converge to the *global solution*, existing for $t \in \mathbb{R}$, because δ_k may become arbitrary small.

Definition 2.2.1 *Assume that $f : G \to \mathbb{R}^n$ is continuous and locally Lipschitz continuous in z on the open set $G \subseteq \mathbb{R}^{n+1}$ and $(t_0, z_0) \in G$. We define the quantities $t_\pm \in \mathbb{R} \cup \{\pm\infty\}$ as follows:*

$$\begin{aligned} t_+(t_0, z_0) :=&\ \sup\{\tau > t_0 : \textit{there exists a extension of (2.1) to } [t_0, \tau]\}, \\ t_-(t_0, z_0) :=&\ \ \inf\{\tau < t_0 : \textit{there exists a extension of (2.1) to } [\tau, t_0]\}. \end{aligned}$$

The interval $[t_-, t_+]$ is called maximal existence interval of the solution of (2.1).
The maximal solution $z(t)$ of the IVP is defined as $z(t) = z_+(t)$, $t \in [t_0, t_+)$, where $z_+(t)$ is the extension of the solution on the interval $[t_0, t_+]$. For $t \in (t_-, t_0)$ the maximal solution is defined as $z(t) = z_-(t)$ where $z_-(t)$ is the extension of the solution on the interval $[t_-, t_0]$.

2.3 Ordinary differential equations with singularities

We now consider singular ODEs of the form

$$z'(t) = \frac{\lambda}{t^\alpha} z(t) + g(t) =: f(t, z(t)), \quad t \in (0, 1], \ \ \alpha \geq 1.$$

We easily see that the right-hand side $f(t, z(t))$ is not continuous in $t = 0$. Also, due to

$$|f(t, z_1(t)) - f(t, z_2(t))| \leq \frac{1}{t^\alpha} |\lambda| \, |z_1(t) - z_2(t)|$$

$f(t, z(t))$ is not Lipschitz continuous in z on the interval $[0, 1]$. Therefore, we cannot use the standard results to describe the existence of continuous solutions of the singular ODE. It turns out that in case of singularities the existence of continuous solutions depends on the sign of λ. This question will be discussed in detail in the following section.

2.4 Analytical results for BVPs with a singularity of the first kind

In this section we first discuss the scalar ODE

$$z'(t) = \frac{\lambda}{t} z(t) + g(t), \quad t \in (0,1], \tag{2.2}$$

where $g \in C[0,1]$ and $\lambda \in \mathbb{R}$ and then generalize the results to a linear system

$$z'(t) = \frac{M}{t} z(t) + g(t), \quad t \in (0,1], \tag{2.3}$$

where M is a $n \times n$ real-valued matrix and $g, z : [0,1] \to \mathbb{R}^n$.

2.4.1 General solution of the homogeneous problem

We first solve the homogeneous problem in (2.2),

$$z'(t) = \frac{\lambda}{t} z(t), \quad t \in (0,1].$$

Clearly, the general solution of the above differential equation is given by

$$z_h(t) = \exp\left(\int_1^t \frac{\lambda}{s} ds \right) c = \mathrm{e}^{\lambda(\ln t - \ln 1)} c = t^\lambda c.$$

2.4.2 Particular solution of the inhomogeneous problem

We use the variation of constant to solve the inhomogeneous problem. Therefore, we make the ansatz

$$z_p(t) = t^\lambda c(t)$$

and substitute $z_p(t)$, into the differential equation (2.2). Consequently, we can calculate the unknown function $c(t)$,

$$\begin{aligned}
\left(t^\lambda c(t)\right)' &= \frac{\lambda}{t} t^\lambda c(t) + g(t) \;\Rightarrow \\
\lambda t^{\lambda-1} c(t) + t^\lambda c'(t) &= \lambda t^{\lambda-1} c(t) + g(t) \;\Rightarrow \\
t^\lambda c'(t) &= g(t) \;\Rightarrow \\
c'(t) &= t^{-\lambda} g(t) \;\Rightarrow \\
c(t) &= \int_1^t s^{-\lambda} g(s)\; ds
\end{aligned}$$

and obtain the general solution of the inhomogeneous ODE as

$$z(t) = z_h(t) + z_p(t) = t^\lambda c + t^\lambda \int_1^t s^{-\lambda} g(s)\; ds.$$

Note that in general $z \in C(0,1]$. In order to construct $z \in C[0,1]$, we have to distinguish between three cases depending on λ.

2.4.3 Continuous and unique solution of the inhomogeneous problem

We now discuss the properties of

$$z(t) = z_h(t) + z_p(t) = t^\lambda c + t^\lambda \int_1^t s^{-\lambda} g(s)\; ds.$$

Case 1: $\lambda < 0$

In this case the above particular solution z_p is not continuous in $t = 0$. To gain conditions for the continuity we split $z(t)$ into two parts,

$$\begin{aligned}
z(t) =& t^\lambda c + t^\lambda \int_1^t s^{-\lambda} g(s)\; ds = t^\lambda \left(c + \int_1^0 s^{-\lambda} g(s)\; ds + \int_0^t s^{-\lambda} g(s)\; ds \right) = \\
=& t^\lambda \tilde{c} + t^\lambda \int_0^t s^{-\lambda} g(s)\; ds.
\end{aligned}$$

In the above integral, we now substitute $u := s/t$ and obtain

$$\tilde{z}_p(t) := t^\lambda \int_0^t s^{-\lambda} g(s)\ ds = t^\lambda \int_0^1 (tu)^{-\lambda} g(tu) t\ du = t \int_0^1 u^{-\lambda} g(tu)\ du \in C[0,1],$$

where $\tilde{z}_p(0) = 0$. Since t^λ is not continuous on the interval $[0,1]$, we have to choose $\tilde{c} = 0$ and so the unique continuous solution of the initial value problem

$$z'(t) = \frac{\lambda}{t} z(t) + g(t), \quad t \in (0,1], \quad z(0) = 0, \tag{2.4}$$

is

$$z(t) = t \int_0^1 u^{-\lambda} g(tu)\ du.$$

To discuss the higher derivatives of z, we substitute z into the differential equation and obtain,

$$z'(t) = \lambda \int\limits_0^1 u^{-\lambda} g(tu)\ du + g(t),$$

and by further differentiation of $z'(t)$, it follows

$$z^{(n+1)}(t) = \lambda \int\limits_0^1 u^{-\lambda+n} g^{(n)}(tu)\ du + g^{(n)}(t).$$

Clearly, for $g \in C^n[0,1]$, $z \in C^{n+1}[0,1]$.

Case 2: $\lambda = 0$

In this case the ODE reduces to $z'(t) = g(t)$ and the solution reads:

$$z(t) = z(1) + \int_1^t g(s)\ ds = z(0) + \int_0^t g(s)\ ds.$$

This solution is continuous on $[0,1]$ and it becomes unique by prescribing its values at $t = 1$ or $t = 0$. Therefore, it holds for the terminal or initial value problem

$$z'(t) = g(t), \quad t \in [0,1], \quad z(1) = \beta, \quad z(t) = \beta + \int_1^t g(s)\ ds, \tag{2.5}$$

$$z'(t) = g(t), \quad t \in [0,1], \quad z(0) = \beta, \quad z(t) = \beta + \int_0^t g(s)\ ds, \tag{2.6}$$

respectively. Moreover, we can see that for $g \in C^n[0,1]$, $z \in C^{n+1}[0,1]$.

Case 3: $\lambda > 0$

We now show that for $\lambda > 0$, the solution given by

$$z(t) = z_h(t) + z_p(t) = t^\lambda c + t^\lambda \int_1^t s^{-\lambda} g(s)\ ds$$

is continuous on $[0,1]$. Clearly $z_h \in C[0,1]$, so we only have to deal with $z_p(t)$. We estimate z_p as follows:

$$|z_p(t)| = t^\lambda \left| \int_1^t s^{-\lambda} g(s)\ ds \right| \leq \|g\|_\infty t^\lambda \left| \int_1^t s^{-\lambda}\ ds \right|.$$

To continue, we distinguish between two cases.

Case 3a: $\lambda \neq 1$

We calculate the integral and have,

$$|z_p(t)| \leq \|g\|_\infty t^\lambda \left| \frac{s^{1-\lambda}}{1-\lambda}\bigg|_1^t \right| = \|g\|_\infty \left| \frac{t^\lambda \left(t^{1-\lambda} - 1\right)}{1-\lambda} \right| = \|g\|_\infty \left| \frac{t - t^\lambda}{1-\lambda} \right|.$$

One can see that $\lim_{t\to 0} z_p(t) = 0$ and $\lim_{t\to 0} z(t) = 0$. This means that $z \in C[0,1]$. Now we derive the formula for the first derivative. We substitute the above solution representation into the differential equation,

$$z'(t) = g(t) + \lambda \left(t^{\lambda-1} c + t^{\lambda-1} \int_1^t s^{-\lambda} g(s)\ ds \right),$$

and it is clear that for $g \in C[0,1]$ and $\lambda > 1$, $z \in C^1[0,1]$. Using integration by parts in the above representation for z', we can rewrite z' and obtain

$$z'(t) = g(t) + \lambda \left(t^{\lambda-1} c + \frac{1}{-\lambda+1} \left(g(t) - t^{\lambda-1} g(1)\right) - \frac{t^{\lambda-1}}{-\lambda+1} \int_1^t s^{-\lambda+1} g'(s)\ ds \right).$$

Taking a derivative of $z'(t)$ yields

$$z''(t) = g'(t) + \lambda\left((\lambda-1)t^{\lambda-2}c + t^{\lambda-2}g(1) + t^{\lambda-2}\int\limits_1^t s^{-\lambda+1}g'(s)\ ds\right)$$

and for $g \in C^1[0,1]$ and $\lambda > 2$, $y \in C^2[0,1]$. Similarly, we can derive the representation for higher derivatives of z and conclude that $z \in C^{n+1}[0,1]$ if $g \in C^n[0,1]$ and $\lambda > n+1$.

Case 3b: $\lambda = 1$

Here, we have

$$z(t) = z_h(t) + z_p(t) = tc + t\int\limits_1^t s^{-\lambda}g(s)\ ds$$

and

$$|z_p(t)| = \left|t\int\limits_1^t \frac{1}{s}g(s)\ ds\right| \le |t|\left|\int\limits_1^t \frac{1}{s}\right|\ ds\,\|g\|_\infty \le |t\ln t|\,\|g\|_\infty$$

and again $z \in C[0,1]$ with $\lim_{t\to0} z(t) = 0$. To obtain the first derivative of z, we substitute the above solution representation into the differential equation,

$$z'(t) = \frac{1}{t}z(t) + g(t) = c + \int\limits_1^t \frac{1}{s}g(s)\ ds + g(t),$$

and it is clear that in general only $z \in C^1(0,1]$ holds.

By means of partial integration we obtain,

$$z(t) = tc + t\int\limits_1^t \frac{1}{s}g(s)\ ds = tc + t\ln(t)g(t) - t\int_1^t \ln(s)g'(s)ds,$$

and the first derivative of z reads:

$$z'(t) = \lambda\left(c + \ln(t)g(t) - \int_1^t \ln(s)g'(s)ds\right) + g(t).$$

In this case, $z \in C[0,1] \cap C^1(0,1]$ in general.

We recapitulate the case $\lambda > 0$. The following terminal value problem (TVP)

$$z'(t) = \frac{\lambda}{t} z(t) + g(t), \quad z(1) = \beta,$$

has for any $g \in C[0,1]$ and $\beta \in \mathbb{R}$ the unique solution $z \in C[0,1]$,

$$z(t) = t^{\lambda}\beta + t^{\lambda} \int_1^t s^{-\lambda} g(s)\, ds.$$

Moreover, $z(0) = 0$ and $z \in C^{n+1}[0,1]$ if $g \in C^n[0,1]$ and $\lambda > n+1$.

This structure of solutions in the scalar case will now be used to describe the solvability of systems. Especially we consider the IVP

$$z'(t) = \frac{M}{t} z(t) + g(t), \quad t \in (0,1], \quad B_0 z(0) = \beta, \tag{2.7}$$

where $M \in \mathbb{R}^{n \times n}$, $B_0 \in \mathbb{R}^{m \times n}$, and $\beta \in \mathbb{R}^m$, $m \leq n$,
and the TVP

$$z'(t) = \frac{M}{t} z(t) + g(t), \quad t \in (0,1], \quad B_1 z(1) = \beta, \tag{2.8}$$

where $M \in \mathbb{R}^{n \times n}$, $B_1 \in \mathbb{R}^{n \times n}$, and $\beta \in \mathbb{R}^n$.

In [7] boundary conditions, which are necessary and sufficient for $z \in C[0,1]$ were formulated, the following lemmas are taken from [17].
To specify them we first decouple the system. Let J be the Jordan canonical form of M and E the associated matrix of generalized eigenvectors such that

$$M = EJE^{-1}.$$

Let

$$v(t) = E^{-1} z(t), \quad \tilde{g}(t) := E^{-1} g(t),$$

then

$$v'(t) = \frac{J}{t} v(t) + \tilde{g}(t), \quad t \in (0,1].$$

For simplicity, let us assume J to be diagonal and all eigenvalues of M be real. Then J takes the form

$$J = \begin{pmatrix} J_- & & \\ & J_0 & \\ & & J_+ \end{pmatrix},$$

where all eigenvalues in J_- are negative, in J_0 zero, and in J_+ positive. We can generalize the results below to any spectrum of M.

Definition 2.4.1 *Let the eigenspace of M associated with positive eigenvalues be denoted by X_+ and the eigenspace of M associated with eigenvalues equals zero by X_0. Moreover, let S is the orthogonal projection onto X_+ and R the orthogonal projection onto X_0. Furthermore, $P := S + R$ is the orthogonal projection onto $X_+ \oplus X_0$ and $Q := I - P$.*

We now make the following assumptions:

A1. For an IVP (2.7) all eigenvalues λ are either $\lambda < 0$ or $\lambda = 0$.

A2. For an TVP (2.8) all eigenvalues λ are either $\lambda > 0$ or $\lambda = 0$.

Lemma 2.4.2 *Let A1 hold and $z \in C[0,1]$ be a general solution of the ODE system (2.7). Then $I = Q + R$ and*

$$Qz(0) = 0, \quad Mz(0) = MRz(0) = 0.$$

This result means that the requirement $z \in C[0,1]$ is equivalent to $\operatorname{rank}(Q) = \operatorname{rank}(M) = n - \operatorname{rank}(R) = n - m$ homogeneous initial conditions z has to satisfy. Note that here $\mathrm{R} = m$.

The next lemma illustrates the solution of the IVP

$$z'(t) = \frac{M}{t} z(t) + g(t), \quad Qz(0) = 0, \quad B_0 z(0) = \beta.$$

Lemma 2.4.3 *Let A1 hold and let the $m \times m$ matrix $B_0\tilde{R}$ be nonsingular. Where $\tilde{R} \in \mathbb{R}^{n\times m}$ is the matrix consisting of the linearly independent columns of R. Then for every $g \in C^p[0,1]$, $p \geq 0$, and any vector $\beta \in \mathbb{R}^m$, there is a unique solution*

$z \in C^{p+1}[0,1]$ of the above IVP. This solution has the form

$$z(t) = \tilde{R}(B_0\tilde{R})^{-1}\beta + t\int_0^1 s^{-M}g(st)\,ds,$$

where $s^{-M} = Es^{-J}E^{-1}$ and

$$s^{-J} = \begin{pmatrix} s^{-\lambda_1} & & & \\ & s^{-\lambda_2} & & \\ & & \ddots & \\ & & & s^{-\lambda_n} \end{pmatrix}, \qquad \text{if} \qquad J = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}.$$

A similar lemma can be formulated for the TVP.

Lemma 2.4.4 *Let A2 hold and let z be a general solution the ODE system in (2.8). Then $I = R + S$, $z \in C[0,1]$, and*

$$Sz(0) = 0.$$

This result means that the smoothness requirement $z \in C[0,1]$ is satisfied by any solution. The following lemma shows the corresponding result for the TVP

$$z'(t) = \frac{M}{t}z(t) + g(t), \quad B_1 z(1) = \beta.$$

Lemma 2.4.5 *Let A2 hold and let the $n \times n$ matrix B_1 be nonsingular. Then for every $g \in C^p[0,1]$, $p \geq 0$, and any vector $\beta \in \mathbb{R}^n$, there exists a unique solution $z \in C[0,1] \cap C^{p+1}(0,1]$ of the above TVP. This solution is given by*

$$z(t) = t^M B_1^{-1}\beta + t^M \int_1^t s^{-M}g(s)\,ds.$$

In case that the smallest positive eigenvalue of M, $\lambda > p+1$, $z \in C^{p+1}[0,1]$.

Finally, let us consider the BVP

$$z'(t) = \frac{M}{t}z(t) + g(t), \quad Qz(0) = 0, \;\; Sz(1) = S\gamma, \;\; Rz(0) = R\gamma.$$

Here the spectrum of M can include negative, zero, and positive eigenvalues. Also $I = Q + S + R$.

Lemma 2.4.6 *For every $g \in C[0,1]$, and every $\gamma \in \mathbb{R}^n$ there exists a unique solution $z \in C[0,1]$ of the above BVP. This solution has the form*

$$z(t) = t^M (S+R)\gamma + (Kg)(t) = t^M P\gamma + (Kg)(t),$$

where $K : C[0,1] \to C[0,1]$,

$$(Kg)(t) = tQ\int_0^1 s^{-M} g(ts)ds + t^M S\int_1^t s^{-M} g(s)ds + tR\int_0^1 s^{-M} g(ts)ds.$$

We now use the previous considerations to formulate analogous result for the general BVP

$$z'(t) = \frac{M}{t} z(t) + g(t), \quad Qz(0) = 0, \;\; B_0 z(0) + B_1 z(1) = \beta. \tag{2.9}$$

Lemma 2.4.7 *Let $\tilde{P} \in \mathbb{R}^{n\times m}$ be the matrix consisting of the linearly independent columns of P. Then $Y(t) = t^M \tilde{P}$ is the unique continuous $n \times m$ matrix satisfying*

$$Y'(t) = \frac{M}{t} Y(t), \quad t \in [0,1], \quad Y(1) = \tilde{P}.$$

Moreover, there exists a unique solution $z \in C[0,1]$ of the BVP (2.9), iff for the matrices B_0, $B_1 \in \mathbb{R}^{m\times n}$ with $m = rank(P)$ and the right hand side $\beta \in \mathbb{R}^m$, the $m \times m$ matrix

$$B_0 R Y(0) + B_1 \tilde{P}$$

is nonsingular.

For proofs and technical details see [7].

3 Collocation Method - `bvpsuite`

In this section, we collect the most important results from [1], [5], and [10].
Here, we focus on the numerical solution of singular boundary value problems of the form

$$
\begin{aligned}
z'(t) &= \frac{M(t)}{t^\alpha} z(t) + f(t, z(t)), \quad t \in (0,1], && (3.1\text{a}) \\
B_0\, z(0) + B_1\, z(1) &= \beta, && (3.1\text{b})
\end{aligned}
$$

where $\alpha \geq 1$, z is a n-dimensional real function, M is a smooth $n \times n$ matrix and f is a n-dimensional smooth function defined on a suitable domain. B_0 and B_1 are constant matrices which are subject to certain restrictions for a well-posed problem.

3.1 Notation

Throughout this chapter, following notations will be used. For functions $z \in C[0,1]$, we define the maximum norm,

$$\|z\| := \max_{0 \leq t \leq 1} |z(t)|,$$

where

$$|z(t)| := \max_{1 \leq k \leq n} |z_k(t)|.$$

On the interval $[0,1]$, we define a mesh

$$\Delta := (\tau_0, \tau_1, \ldots, \tau_N), \quad \tau_0 = 0,\ \tau_N = 1,$$

such that

$$h_i := \tau_{i+1} - \tau_i, \quad J_i := [\tau_i, \tau_{i+1}], \quad i = 0, \ldots, N-1.$$

For reasons of simplicity, we restrict the discussion to equidistant meshes,

$$h_i = h, \quad i = 0, \dots, N-1.$$

However, the results also hold for nonuniform meshes, which have a limited variation in the stepsizes, cf. [8]. On Δ, we define corresponding grid vectors

$$u_\Delta := (u_0, \dots, u_N) \in \mathbb{R}^{(N+1)n}. \tag{3.2}$$

The norm on the space of grid vectors is given by

$$\|u_\Delta\|_\Delta := \max_{0 \le k \le N} |u_k|. \tag{3.3}$$

For a continuous function $z \in C[0,1]$, we denote by R_Δ the pointwise projection onto the space of grid vectors,

$$R_\Delta(z) := (z(\tau_0), \dots, z(\tau_N)). \tag{3.4}$$

For the collocation, m points $t_{i,j},\ j = 1, \dots, m$, are inserted in each subinterval J_i. We choose the same distribution of collocation points in every subinterval, thus yielding the (fine) grid[1]

$$\Delta^m = \Delta \cup \{t_{i,j} = \tau_i + \rho_j h,\ \ i = 0, \dots, N-1,\ \ j = 1, \dots, m\},$$

with

$$0 < \rho_1 < \rho_2 \cdots < \rho_m \le 1.$$

We choose the grids where $\rho_1 > 0$ to avoid a special treatment of the singular point $t = 0$ [3].

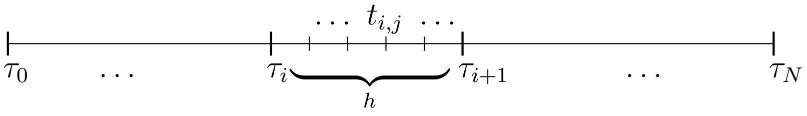


Figure 3.1: The computational grid.

For a grid Δ^m, u_{Δ^m}, $\|\cdot\|_{\Delta^m}$ and R_{Δ^m} are defined analogously to (3.2)–(3.4).

[1] For convenience, we denote τ_i by $t_{i,0} \equiv t_{i-1,m+1},\ i = 1, \dots, N$.

3.2 Collocation Method

Let us choose Δ and Δ^m as described in 3.1 and donate by B_m the Banach space of globally continuous piecewise polynomial functions of degree $\leq$ m, equipped with the norm $\|\cdot\|_\Delta$. The idea of the collocation method is to approximates the solution z of (3.1) by a function $P \in B_m$ which satisfies the collocation conditions

$$P'(t_{i,j}) = \frac{M(t_{i,j})}{t_{i,j}^\alpha} P(t_{i,j}) + f(t_{i,j}, P(t_{i,j})), \quad i = 0, \dots, N-1, \; j = 1, \dots, m, \tag{3.5a}$$

subject to boundary conditions

$$B_0 P(0) + B_1 P(1) = \beta. \tag{3.5b}$$

3.2.1 Convergence

In [12] the following convergence result for differential equations with a singularity of the first kind, $\alpha = 1$, was proven.

Theorem 3.2.1 *Assume that $M \in C^{m+2}[0,1]$, f is $m+1$ times continuously differentiable in $[0,1] \times \mathbb{R}^n$ with $\frac{\partial f}{\partial z}$ bounded in that domain and for σ_+, the smallest of the positive real parts of the eigenvalues of $M(0)$, holds $\sigma_+ > m+2$. Then the collocation scheme (3.5) has a unique solution $P \in B_m$ in a neighborhood of an isolated solution $z \in C^{m+2}[0,1]$ of (3.1). This solution can be computed using Newton's method, which converges quadratically. Moreover,*

$$\begin{aligned}
&\|P - z\| = O(h^m), \\
&\left| \frac{M(0)}{t}(P(t) - z(t)) \right| = O(h^m), \quad t \in [0,1], \\
&\left\| P^{(k+1)} - z^{(k+1)} \right\| = O(h^{m-k}), \quad k = 0, \dots, m-1, \\
&\left| P'(t) - \frac{M(t)}{t} P(t) - f(t, P(t)) \right| = O(h^m), \quad t \in [0,1].
\end{aligned}$$

Note that the condition $\sigma_+ > m+2$ does not impose a restriction of generality, see [10] for further details.

For ODEs with an essential singularity, $\alpha > 0$, no corresponding analytical result is known. Though, the stage order $O(h^m)$ can be seen in experiments for any choice of symmetric collocation points.

3.2.2 Basic Solver in the Matlab Code bvpsuite

The code is designed to solve systems of differential equations of arbitrary mixed order including zero[2], subject to initial or boundary conditions,

$$F\big(t, p_1, \dots, p_s, z_1(t), z_1'(t), \dots, z_1^{(l_1)}(t), \dots, z_n(t), z_n'(t), \dots, z_n^{(l_n)}(t)\big) = 0, \tag{3.6a}$$

$$\begin{aligned} B\big(&p_1, \dots, p_s, z_1(c_1), \dots, z_1^{(l_1-1)}(c_1), \dots, z_n(c_1), \dots, z_n^{(l_n-1)}(c_1), \dots, \\ &z_1(c_q), \dots, z_1^{(l_1-1)}(c_q), \dots, z_n(c_q), \dots, z_n^{(l_n-1)}(c_q)\big) = 0, \end{aligned} \tag{3.6b}$$

where the solution $z(t) = (z_1(t), z_2(t), \dots, z_n(t))^T$, and the parameters p_i, $i = 1, \dots, s$, are unknown. In general, $t \in [a, b], -\infty < a, b < \infty$[3]. Moreover,

$$F : [a, b] \times \mathbb{R}^s \times \mathbb{R}^{l_1+1} \times \dots \times \mathbb{R}^{l_n+1} \to \mathbb{R}^n$$

and

$$B : \mathbb{R}^s \times \mathbb{R}^{ql_1} \times \dots \times \mathbb{R}^{ql_n} \to \mathbb{R}^{l+s},$$

where $l := \sum_{k=1}^n l_k$. Note that boundary conditions can be posed on any subset of distinct points $c_i \in [a, b]$, $a \le c_1 < c_2 < \dots < c_q \le b$. To find an numerical solution for (3.6) we search for a piecewise polynomial function $P \in B_m$, where $P(t) := P_i(t)$, $t \in J_i$, see Figure 3.2. Since every subinterval contains m collocations points, the k-th component of P_i is a polynomial of degree smaller or equal $m + l_k - 1$, $k = 1, \dots, n$. Let us now formulate all equations P has to satisfy. First set are the collocation conditions which mean that the differential equation is satisfied exactly (up to the round-off errors)

[2] This means that differential-algebraic equations are also in the scope of the code.

[3] The code can also deal with problems posed on semi-infinite intervals $t \in (a, \infty)$, $a > 0$ (and by a splitting of the interval, also with $a = 0$). In order to exploit the efficient and robust mesh selection strategy, we use the transformation $t = \frac{1}{\tau}$, $z(t) = x\left(\frac{1}{\tau}\right)$ to restate

$$x'(\tau) = \tau^\beta f(\tau, x(\tau)), \quad \tau \in [a, \infty), \quad \beta > -1$$

as

$$z'(t) = -\frac{1}{t^{\beta+2}} f(1/t, z(t)), \quad t \in (0, 1/a].$$

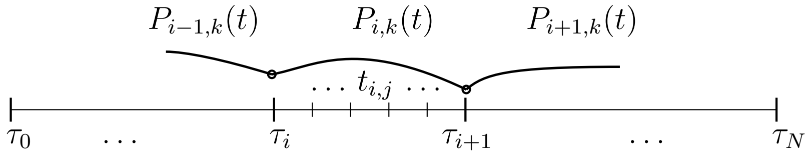


Figure 3.2: The collocation method for one solution component.

in the collocation points,

$$\begin{aligned}&F\big(t_{i,j}, p_1, \dots, p_s, P_{i,1}(t_{i,j}), P'_{i,1}(t_{i,j}), \dots, P^{(l_1)}_{i,1}(t_{i,j}), \dots, \\ &\qquad P_{i,n}(t_{i,j}), P'_{i,n}(t_{i,j}), \dots, P^{(l_n)}_{i,n}(t_{i,j})\big) = 0, \quad i = 0, \cdots N-1,\ j = 1, \dots, m.\end{aligned} \tag{3.7a}$$

Additionally, we have the boundary condition, here formulated only for a two-point BVP,

$$\begin{aligned}&B\big(p_1, \dots, p_s, P_{0,1}(0), \dots, P^{(l_1-1)}_{0,1}(0), \dots, P_{0,n}(0), \dots, P^{(l_n-1)}_{0,n}(0), \dots, \\ &\qquad P_{N-1,1}(1), \dots, P^{(l_1-1)}_{N-1,1}(1), \dots, P_{N-1,n}(1), \dots, P^{(l_n-1)}_{N-1,n}(1)\big) = 0,\end{aligned} \tag{3.7b}$$

and the continuity requirements

$$P^{(\nu)}_{i,k}(\tau_{i+1}) = P^{(\nu)}_{i+1,k}(\tau_{i+1}), \quad i = 0, \dots, N-2,\ \nu = 1, \dots, l_k,\ k = 1, \dots, n. \tag{3.7c}$$

Before solving the resulting nonlinear algebraic system of equations, we check if the number of conditions is summing up to the number of unknown coefficients in the polynomial representation. There are N polynomials, each with n components and for each component there are $m + l_k$ unknown coefficients. Together with the s unknown parameters, we have $N(nm + l) + s$ unknowns.
On the other hand (3.7a) provides Nmn, (3.7b) l and (3.7c) $(N-1)\sum l_i = (N-1)l$ equations. This gives $Nmn + l + (N-1)l + s = N(mn + l) + s$ equations.

In bvpsuite, the Runge-Kutta basis is used for the representation of the collocation polynomials. This basis is specified in the following section.

3.2.3 Runge-Kutta basis

A definition of the Runge-Kutta basis for the first order problems in ODEs is given in [1], for the second order problems in [9]. We will describe a basis for first order problems on the interval $[0,1]$ and generalize this to an arbitrary interval $[a,b]$.

Runge-Kutta basis on the interval $[0,1]$

We consider m collocation points on $[0,1]$ given by ρ_j, $j=1,...,m$, $0<\rho_1<\cdots<\rho_m<1$ on the interval $[0,1]$. The $m+1$ Runge-Kutta basis-elements $\varphi_1(t),\Psi_1(t),\dots,\Psi_m(t)$ are defined via following conditions:

$$\varphi_1(t)=1, \quad \Psi_k'(\rho_j)=\delta_{kj}, \quad k,j=1,\dots,m$$

or

$$\begin{pmatrix} \varphi_1(0) & \varphi_1'(\rho_1) & \cdots & \varphi_1'(\rho_m) \\ \Psi_1(0) & \Psi_1'(\rho_1) & \cdots & \Psi_1'(\rho_m) \\ \Psi_2(0) & \Psi_2'(\rho_1) & \cdots & \Psi_2'(\rho_m) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_m(0) & \Psi_m'(\rho_1) & \cdots & \Psi_m'(\rho_m) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

A polynomial represented in Runge-Kutta basis has the form

$$P(t) := y_1\varphi_1(t) + \sum_{j=1}^{m} z_j\Psi_j(t),$$

where the coefficients y_1 and z_j can be expressed as follows

$$y_1 = P(0), \quad z_j = P'(\rho_j).$$

Calculation of the Runge-Kutta basis on the interval $[0,1]$

In [9] an important relation between the Runge-Kutta basis and the monomial basis is given which we now recapitulate. Let

$$\Psi_j(t) = \int_0^t L_j(u)du,$$

where $L_j(t)$ is the j-th Lagrange basis polynomial for the sampling points $\rho_1, \dots, \rho_m$

$$L_j(t) := \prod_{\substack{s=1\\s\neq j}}^{m} \frac{t-\rho_s}{\rho_j-\rho_s}. \tag{3.8}$$

Let $[t^k]P(t)$ denote the coefficient of t^k of the polynomial $P(t)$ in monomial basis. Define

$$c_n^k := [t^k]\prod_{s=1}^{n}(t-a_s), \quad k \leq n,$$

which can be calculated from the following recurrence

$$c_n^k = \begin{cases} c_0^{n-1}(-a_n), & k=0,\\ c_{k-1}^{n-1}+c_k^{n-1}(-a_n), & 1<k<n,\\ 1, & k=n. \end{cases}$$

Consequently, we can write (3.8) in the following form

$$L_j(t) := \frac{\sum\limits_{s=0}^{m-1} c_s^{m-1} t^s}{\prod\limits_{\substack{s=1,\\s\neq j}}^{m}(\rho_j-\rho_s)},$$

with

$$a_s := \begin{cases} \rho_s, & s<j,\\ \rho_{s+1}, & s\geq j. \end{cases}$$

Since $\Psi_j(0)=0$, we obtain

$$\Psi_j'(t) = L_j(t) = \frac{\sum\limits_{s=0}^{m-1} c_s^{m-1} t^s}{\prod\limits_{\substack{s=1,\\s\neq j}}^{m}(\rho_j-\rho_s)},$$

$$\Psi_j(t) = \int_0^t L_j(u)du = \frac{\sum\limits_{s=1}^{m} c_{s-1}^{m-1} \frac{t^s}{s}}{\prod\limits_{\substack{s=1,\\s\neq j}}^{m}(\rho_j-\rho_s)}.$$

Runge-Kutta basis on an arbitrary interval

As defined in Section 3.1, we have in each subinterval $J_i = [\tau_i, \tau_{i-1}]$ the same distribution of the collocation points,

$$t_{i,j} = \tau_i + \rho_j h \in J_i, \quad i = 0, \dots, N-1,\ j = 1, \dots, m.$$

We now define the $m+1$ Runge-Kutta basis polynomials on J_i,

$$B_{RK}^{(i)} := \{\Phi_{i1}(t), \Phi_{i2}(t), \dots, \Phi_{i(m+1)}(t)\}$$

as follows:

$$\begin{aligned} &\Phi_{i1}(t) = 1, \\ &\Phi_{ik}(t) = h\Psi_{k-1}\left(\frac{t-\tau_i}{h}\right), \quad k = 2, \dots, m+1. \end{aligned}$$

Now, we construct a polynomial P_i,

$$P_i(t) = \sum_{j=1}^{m+1} z_{ij}\Phi_{ij}(t) = z_{i1} + h\sum_{j=2}^{m+1} z_{ij}\Psi_{j-1}\left(\frac{t-\tau_i}{h}\right). \tag{3.9}$$

The advantage of this basis is that it is only necessary to calculate and store Ψ_k for the interval $[0,1]$. It is easy to see that the values $\Psi_k'(\frac{t-\tau_i}{h})$ for the sampling points $t = t_{i,1}, \dots, t_{i,m} \in [\tau_i, \tau_{i+1}]$ are identical with the values of $\Psi_k'(\rho)$ for $\rho = \rho_1, \dots \rho_m \in [0,1]$. From

$$t_{ij} = \tau_i + \rho_j(\tau_{i+1} - \tau_i) = \tau_i + \rho_j h,$$

it follows

$$\Psi_k'\left(\frac{t_{ij}-\tau_i}{h}\right) = \Psi_k'\left(\frac{\tau_i + \rho_j h - \tau_i}{h}\right) = \Psi_k'(\rho_j) = \delta_{kj}.$$

Due to

$$\frac{d}{dx}\Psi_k\left(\frac{x-\tau_i}{h}\right) = \frac{1}{h}\frac{d}{d\rho}\Psi_k(\rho)\bigg|_{\rho=\frac{x-\tau_i}{h}}, \quad k = 1, \cdots, m,$$

we obtain from (3.9),

$$z_{i1} = P_i(\tau_i), \quad z_{i(j+1)} = P_i'(t_{ij}), \quad j = 1, \cdots, m.$$

Finally we can write (3.9) in the following way,

$$P_i(t) = P_i(\tau_i) + h \sum_{j=2}^{m+1} P_i'(t_{ij}) \Psi_{j-1} \left(\frac{t - \tau_i}{h} \right), \quad i = 0, \cdots, N-1.$$

The major question which now has to be addressed is the convergence of the scheme for $h \to 0$. This means that we are interested in the behavior of the maximal global error $\|z - P\|_\infty := \max_{0 \le t \le 1} |z(t) - P(t)|$ for $h \to 0$. In particular, it is interesting to know how fast this error decreases, or equivalently, for what $p > 0$ the following statement (a priori error estimate) holds:

$$\|z - P\|_\infty = \max_{0 \le t \le 1} |z(t) - P(t)| = c(h, z) h^p. \tag{3.10}$$

Here, $c(h, z)$ depends on higher derivatives of z and $\lim_{h \to 0} c(h, z) = c > 0$. The constant p is the convergence order of the collocation scheme. Clearly, the representation (3.10) makes sense, when all necessary higher derivatives of z which occur in $c(h, z)$ exist and are bounded on $[0, 1]$. This question of convergence addressed above, has been answered in [4] and it turns out that for an appropriately smooth problem (3.1a)–(3.1b) with a smooth solution z, the convergence order of the collocation scheme is $p = m$. This result means that for problems with smooth solutions it is more efficient to use high order methods than the low order ones, especially when the global error shall be small. To see this, let us assume that all solution derivatives are moderate, $c(h, z) = O(1)$. Then $\|z - P\|_\infty \approx h^p$. Further assume that we wish $\|z - P\|_\infty \approx 10^{-7}$. For a low order method, with for example $p = 1$, we have to use the stepsize $h \approx 10^{-7}$, while for a method of order $p = 7$ it is sufficient to use $h \approx 10^{-1}$. Consequently, in the first case we have to solve for around $2 \cdot 10^7$ unknowns[4], while in the second case the number of unknowns is around 80.

We see that collocation provides an approximation P for the solution z on a prescribed grid Δ_h, where the step size may be constant or vary. In general, a software package for solving BVPs in ODEs provides additional modules controlling the computational

[4] Since $m = p = 1$, we work with polynomials P_i of degree 1 and hence, each of P_i is uniquely specified by 2 unknown parameters. The stepsize $h = 10^{-7}$ means that on the interval $[0, 1]$ we have to compute 10^7 polynomials. Therefore, in this case the number of unknowns is $2 \cdot 10^7$.

process. We now motivate and discuss these controlling mechanisms – error estimate and grid adaptation procedures – in some detail.

3.2.4 Error Estimates for the Global Error of the Collocation

The estimation of the error of P is necessary, because the user not only specifies the problem and expects to obtain an approximation for its solution, but also prescribes how accurate the numerical solution shall be. Using a tolerance parameter TOL, the user may wish the maximal error of the approximation to satisfy $\|z - P\|_\infty \leq TOL$. This means that we have to compute an estimate est for the unknown global error of the collocation polynomial, $z - P$, in order to be able to check if the requirement $\|est\|_\infty \leq TOL$ is satisfied. If this tolerance requirement was not satisfied on a grid Δ_h, we can half the step size and try the grid $\Delta_{h/2}$. Since, according to representation (3.10), decreasing the step size results in decreasing the global error (until the round off error level is reached), we shall be able to find h which is sufficiently small for the approximation to become appropriately precise. Clearly, the error estimate has to reflect the size of the true error correctly, at least for fine grids with small h. Error estimate satisfying this property is called asymptotically correct.

To provide an asymptotically correct estimate for the global error of the collocation solution, we propose to use the classical error estimate based on mesh halving. In this approach, we compute the collocation solution on a mesh Δ_h with the step size h and denote this approximation by $P_{\Delta_h}(t)$. Subsequently, we choose a second mesh $\Delta_{h/2}$ where in every interval of Δ_h we insert two subintervals of length $h/2$. On this mesh, we compute the numerical solution using the same collocation scheme to obtain the collocation polynomial $P_{\Delta_{h/2}}(t)$. Using these two quantities, we define

$$est(t) := 2^m \frac{P_{\Delta_{h/2}}(t) - P_{\Delta_h}(t)}{1 - 2^m}$$

as an error estimate for the approximation $P_{\Delta_h}(t)$. This formula is executed on each subinterval J_i of Δ_h. Generally, estimates of the global error based on mesh halving are robust and therefore, this strategy has been implemented in `bvpsuite`. Note, that this strategy will work analogously for variable step sizes $h_i := \tau_{i+1} - \tau_i$.

3.2.5 Adaptive Mesh Selection

By decreasing the step size coherently, $h \to h/2 \to h/4 \ldots$, it will be in general, possible to satisfy the tolerance requirement but this procedure is inefficient, because it does not take into account the solution behavior and the structure of the error. In Figures 3.3 and 3.4, the advantage of an adaptive grid is illustrated. The underlying analytical problem is a BVP for a system of two equations of the form (3.1a) whose first solution component shows a steep layer at the left end of the interval of integration. We see, that the grid points in the adapted grid very well reflect the solution behavior. With the same number of equidistantly spaced grid points, the same effort is paid but the obtained approximation is unacceptable.

A correct error estimate of the global error is a good indicator for the regions where the solution is difficult to approximate. These regions are usually characterized by a rapid solution change, or equivalently, by large values of its higher derivatives. This also means that the function $c(h, z)$ will be large and so will be the global error. The main idea is to locate the grid points in such a way that the global error becomes equidistributed or constant along the grid. With other words, the grid becomes finer with smaller step sizes in regions where the error is large (solution changes rapidly) and stays coarse with larger step sizes in regions where the error is small (solution changes slowly). This idea can be realized in several ways. The mesh selection strategy discussed below was proposed and investigated in [15]. The new control algorithm consists of two phases. In the first phase carried out on the control grid with a moderate number of points, the grid points are located in such a way that they correctly reflect the solution behavior, cf. Figure 3.3. Most modern mesh generation techniques in two-point boundary value problems construct a smooth function mapping a uniform auxiliary grid ξ to the desired nonuniform grid x. The aim is to construct a grid deformation $\xi = \Phi(x)$ with $\Phi'(x) = \phi(x)$, cf. Figure 3.5. Then $\mathrm{d}\xi = \phi(x)\mathrm{d}x$ and $\Delta\xi \approx \phi(x)\Delta x$. If $\Delta\xi$ is constant then $\Delta x_{n+1/2} = \Delta\xi/\phi(x_{n+1/2})$ varies with $\phi(x)$. Note that ϕ represents the *density* of the grid points – when ϕ is small Δx is large and vice versa. Using an error estimate, a feedback control law generates a new density from the previous one.

In the second phase of the grid adaptation procedure, appropriate number of grid points is added (along the grid density function) to satisfy the tolerance. In Figure 3.6, it can be seen how this strategy works in practice. In the top graph the behavior of the

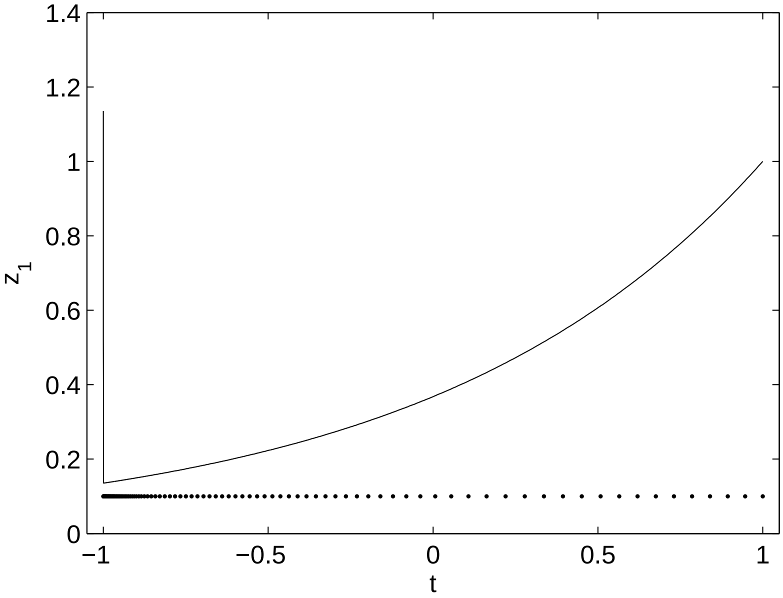


Figure 3.3: Numerical solution and the related adapted grid: $TOL = 10^{-6}$, number of grid points 101, $m = 8$.

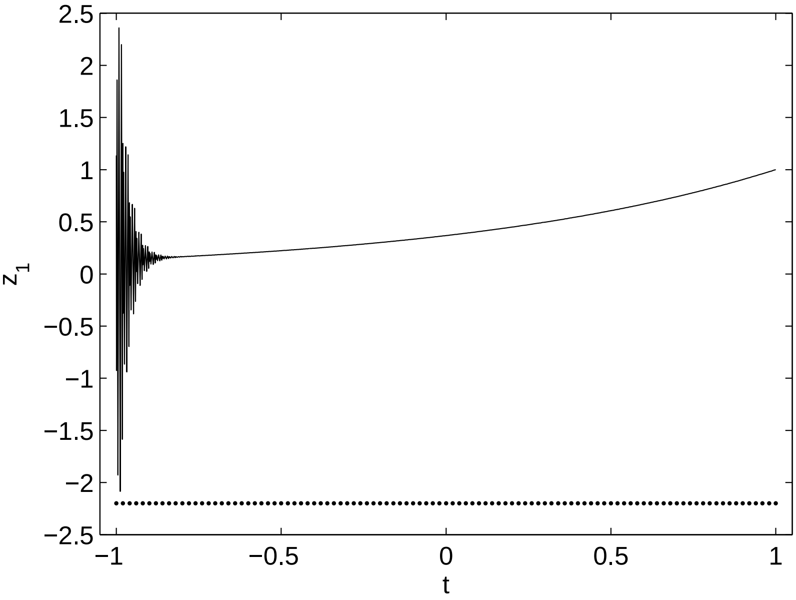


Figure 3.4: Numerical solution and the related uniform grid, number of grid points 101, $m = 8$.

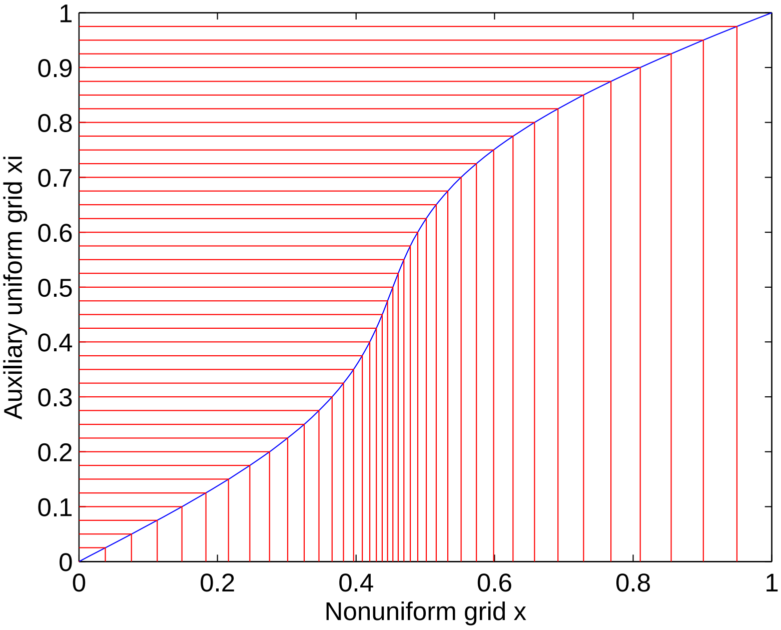


Figure 3.5: Uniform auxiliary grid maps to nonuniform grid, where $x_n = \Phi^{-1}(\xi_n)$.

analytical solution is shown and it is clear that the grid has to be denser in the right part of the interval. In the center graph the grid adaptation procedure is visualized. The control grid consist of 21 points and is equidistant at the beginning. Then, in three iteration steps, the proper location of the grid points in the control grid is found. Finally, on the last grid containing 97 points the tolerance has been satisfied. The bottom graph shows the procedure started on a control grid with 51 points.

We would like to mention that the scope of `bvpsuite` is not restricted to models (3.1a)–(3.1b). The code can cope with fully implicit ODEs of variable order posed on finite or semi-infinite intervals. Differential algebraic equations [13] and parameter-dependent problems [11] are further applications for which `bvpsuite` can be utilized .

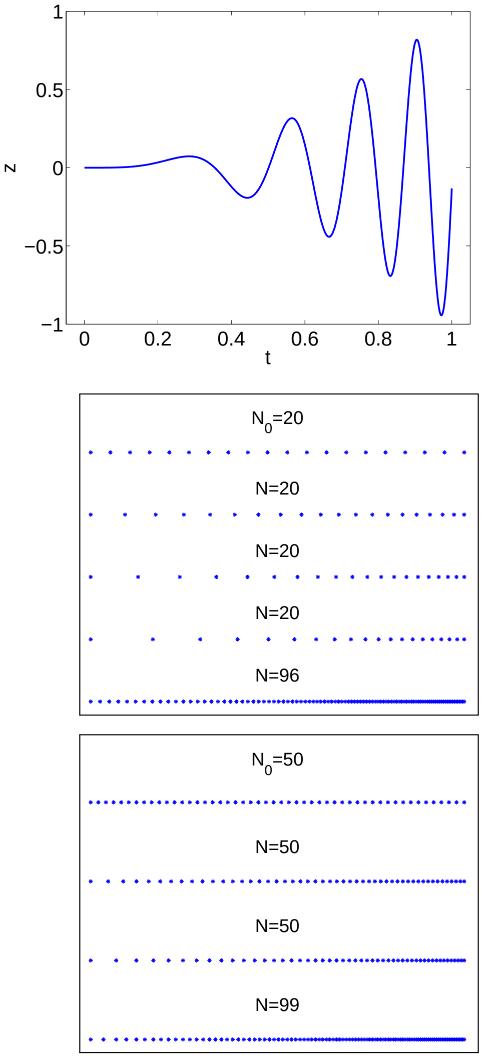


Figure 3.6: Exact solution (upper graph) and the grid adaptation (lower graphs) of `bvpsuite` with collocation of order $m = 4$, $TOL = 10^{-6}$ an initial number of subinterval $N_0 = 20$ and $N_0 = 50$.

4 Applications

4.1 Periodic BVPs in ODEs with time singularities

4.1.1 Problem definition

In paper [6] the existence of solutions to a nonlinear singular second order ordinary differential equation,

$$u''(t) = \frac{a}{t}u'(t) + \lambda f(t, u(t), u'(t)), \quad t \in (0, T)\,, \tag{4.1a}$$

subject to periodic boundary conditions

$$u(0) = u(T),\ u'(0) = u'(T) \tag{4.1b}$$

has been discussed.
Let f satisfy following conditions:

(A_1): $f(\cdot, x, y) : [0, T] \to \mathbb{R}$ is measurable for all $(x, y) \in \mathbb{R}^2$ and $f(t, \cdot, \cdot) : \mathbb{R}^2 \to \mathbb{R}$ is continuous for a.e. $t \in [0, T]$.

(A_2): For a.e. $t \in [0, T]$ and all $(x, y) \in \mathbb{R}^2$ the estimate

$$|f(t, x, y)| \leq g(t)w(|y|)$$

holds with positive functions $g \in L_1[0, T]$ and $w(y) \in C[0, \infty)$, where w is nondecreasing.

Under these assumptions the paper provides the following existence result.

Lemma 4.1.1 *Let $a > 0$. Let conditions (A_1) and (A_2) hold. Assume that there exist*

$A, B \in \mathbb{R}$, such that $A < B$ and

$$f(t,x,y) > 0,\ \textit{for a.e. } t \in [0,T] \textit{ and all } x \le A,\ y \in \mathbb{R},$$
$$\textit{and } f(t,x,y) < 0,\ \textit{for a.e. } t \in [0,T] \textit{ and all } x \ge B,\ y \in \mathbb{R}.$$

Let

$$\lambda^* = \int_0^\infty \frac{ds}{w(s)} \cdot \left(\int_0^T g(t)\, dt \right)^{-1}.$$

Then problem (4.1) has a solution for each $\lambda \in (0, \lambda^*)$.

The circumstance that the lower bound is greater than the upper bound is called the problem *has the opposite-ordered upper and lower functions* We illustrate the solution structure by means of two model problems for the class (4.1).

4.1.2 Example 1

We first examine the boundary value problem,

$$u''(t) = \frac{a}{t}u'(t) + \frac{t}{3} - \frac{(1+u'(t))^2}{4\sqrt{t}} \arctan u(t),\ u(0) = u(1),\ u'(0) = u'(1) \tag{4.2}$$

with $a = 0.4$, 0.5, 0.7, 0.9, 1, 2, and 5.

Figures 4.1 and 4.2 show the numerical results for ODE (4.2) obtained by using 4 collocation points and a uniform mesh of 100 grid points in the interval $[0, 1]$. From Figure 4.2 it is clear that the solutions become more difficult to recover when a decreases. This can also be seen in Figure 4.3 showing the global errors of the approximation.

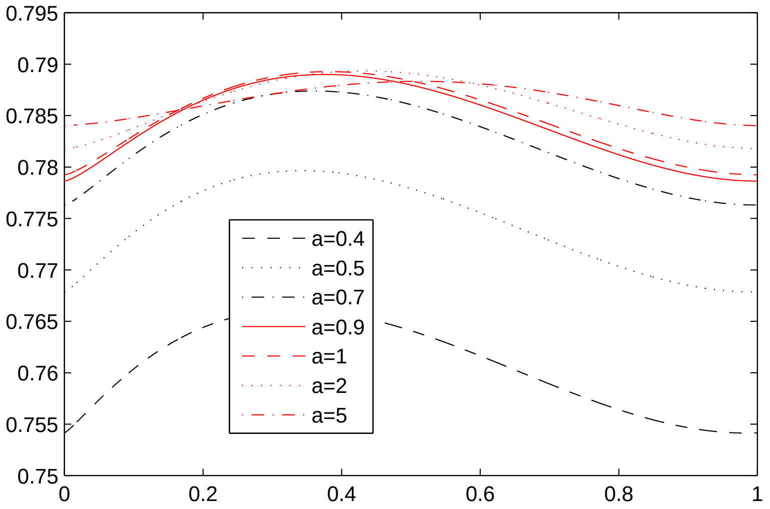


Figure 4.1: Example 1: Numerical solution for different values of a

The analytical solution satisfies $u'(0) = u'(1) = 0$. Therefore, in Figures 4.4 and 4.5 we take a closer look at the regions $t = 0$ and $t = 1$, respectively. We can see that the greater the values of a are, the smaller are the values of the first derivative at $t = 1$. We now try to justify this fact as follows.

Let us look at the differential equation in (4.2). For large values of a, the first term in the right-hand side becomes dominant and we relate the solution smoothness to the linear equation of the form

$$y'(t) = \frac{a}{t}y(t),$$

solved by $y(t) = ct^a$. This immediately explains why the higher derivatives of the solution are smoother for large values of a and consequently, why such solutions are easier to approximate [6, p. 16, 17].

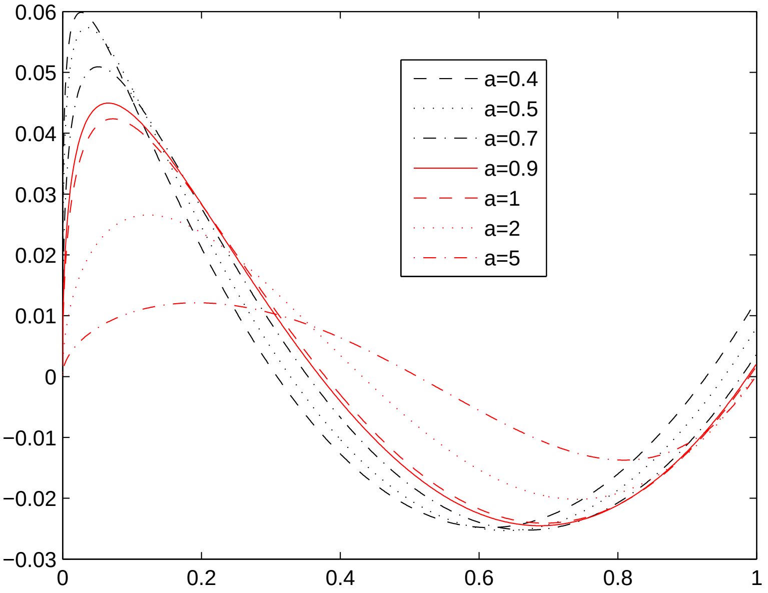


Figure 4.2: Example 1: Numerical values of the first derivative for different values of a

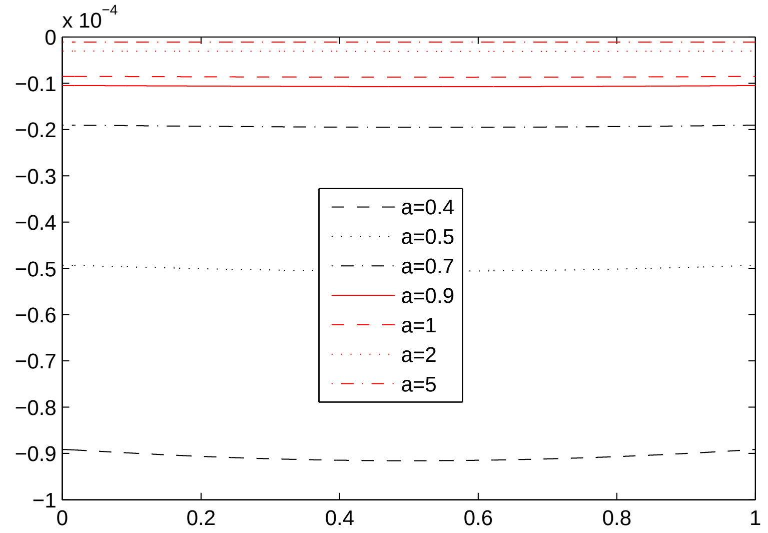


Figure 4.3: Example 1: Global errors for different values of a

a	$u(0) = u(1)$	$u'(0) = u'(1)$	$u''(0)$	maximal error in u
0.4	0.6663581	$1.022613 \cdot 10^{-2}$	$6.047415 \cdot 10^{1}$	$1.42 \cdot 10^{-3}$
0.5	0.7062183	$6.203709 \cdot 10^{-3}$	$4.973534 \cdot 10^{1}$	$1.13 \cdot 10^{-3}$
0.7	0.7455672	$2.709670 \cdot 10^{-3}$	$3.147486 \cdot 10^{1}$	$6.44 \cdot 10^{-4}$
0.9	0.7606680	$1.525785 \cdot 10^{-3}$	$2.065479 \cdot 10^{1}$	$3.94 \cdot 10^{-4}$
1	0.7646459	$1.232963 \cdot 10^{-3}$	$1.722183 \cdot 10^{1}$	$3.23 \cdot 10^{-4}$
2	0.7767797	$4.198916 \cdot 10^{-4}$	$5.939505 \cdot 10^{0}$	$1.12 \cdot 10^{-4}$
5	0.7823136	$1.439668 \cdot 10^{-4}$	$1.947079 \cdot 10^{0}$	$3.86 \cdot 10^{-5}$

Table 4.1: Example 1: Numerical values of $u(0)$, $u'(0)$, $u''(0)$, $u'(1)$, and the maximal global error in u

4.1.3 Example 2

As a second example we consider the boundary value problem

$$u''(t) = \frac{a}{t}u'(t) + \frac{1}{6}\sin(5t) - \frac{1}{5\sqrt[3]{t}}\frac{u(t)(1+|u'(t)|)^3}{\sqrt{1+u(t)^2}}, \quad u(0) = u(1), u'(0) = u'(1)$$

with $a = 0.4$, 0.5, 0.7, 0.9, 1, 2, and 5.

Figures 4.6-4.10 and Table 4.2 correspond to Figures 4.1-4.5 and Table 4.1, respectively.

a	$u(0) = u(1)$	$u'(0) = u'(1)$	$u''(0) = u''(1)$	maximal error in u
0.4	-0.1163804	$-1.344652 \cdot 10^{-3}$	$-3.445983 \cdot 10^{1}$	$1.80 \cdot 10^{-4}$
0.5	-0.1232172	$-5.484997 \cdot 10^{-4}$	$-2.016584 \cdot 10^{1}$	$8.95 \cdot 10^{-5}$
0.7	-0.1272538	$-1.059086 \cdot 10^{-4}$	$-6.832530 \cdot 10^{0}$	$2.16 \cdot 10^{-5}$
0.9	-0.1281526	$-3.320831 \cdot 10^{-5}$	$-2.734550 \cdot 10^{0}$	$7.10 \cdot 10^{-6}$
1	-0.1283698	$-2.308953 \cdot 10^{-5}$	$-1.909115 \cdot 10^{0}$	$4.92 \cdot 10^{-6}$
2	-0.1295174	$-6.364872 \cdot 10^{-6}$	$-4.225124 \cdot 10^{-1}$	$1.35 \cdot 10^{-6}$
5	-0.1309907	$-2.168133 \cdot 10^{-6}$	$-1.271241 \cdot 10^{-1}$	$4.60 \cdot 10^{-6}$

Table 4.2: Example 2: Numerical values of $u(0)$, $u'(0)$, $u''(0)$, $u'(1)$, and the maximal global error in u

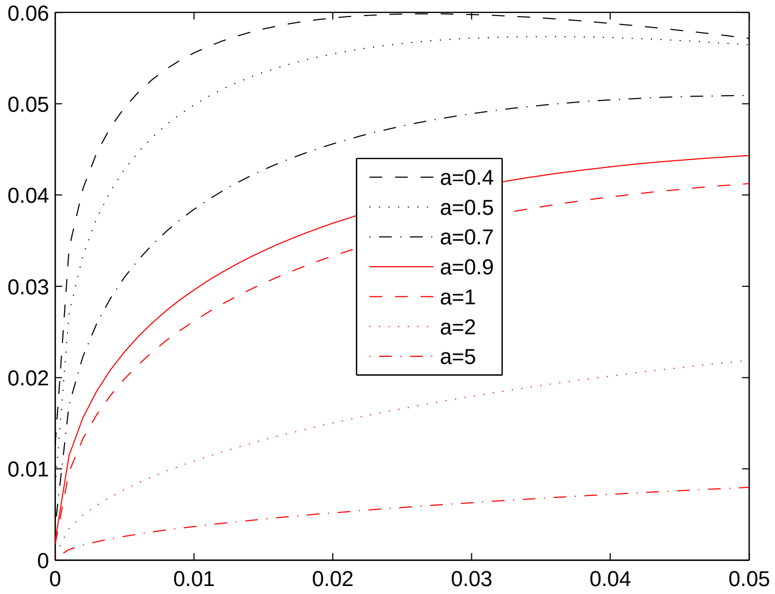


Figure 4.4: Example 1: Numerical values of the first derivative for different values of *a*

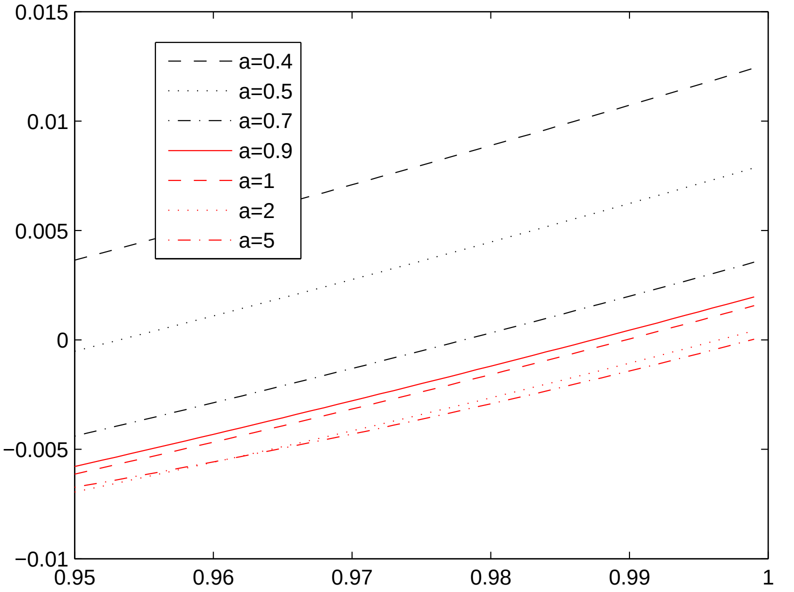


Figure 4.5: Example 1: Global errors for different values of *a*

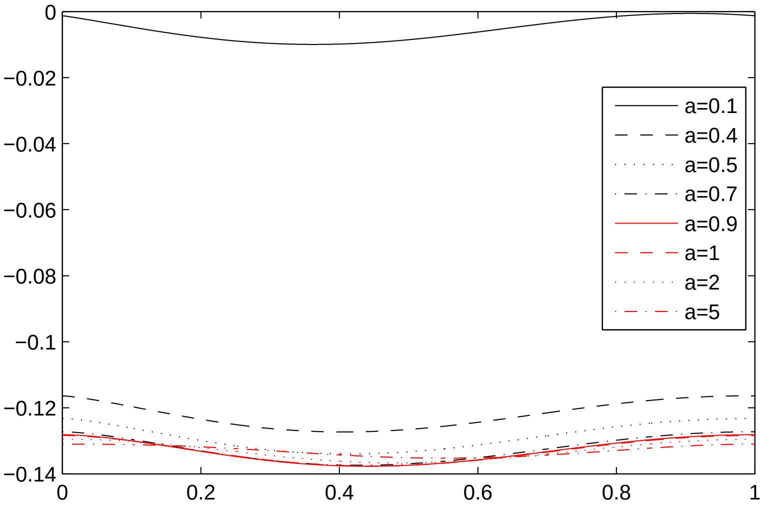


Figure 4.6: Example 2: Numerical solutions for different values of a

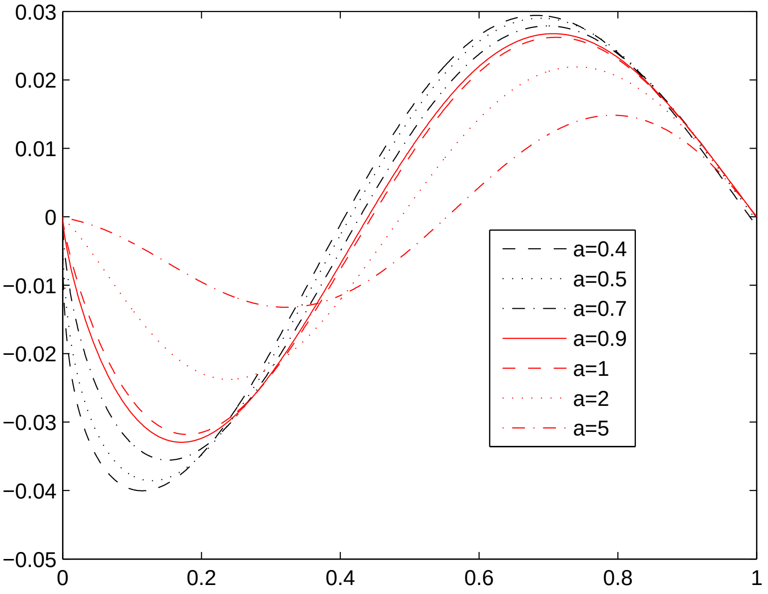


Figure 4.7: Example 2: Numerical values of the first derivative for different values of a

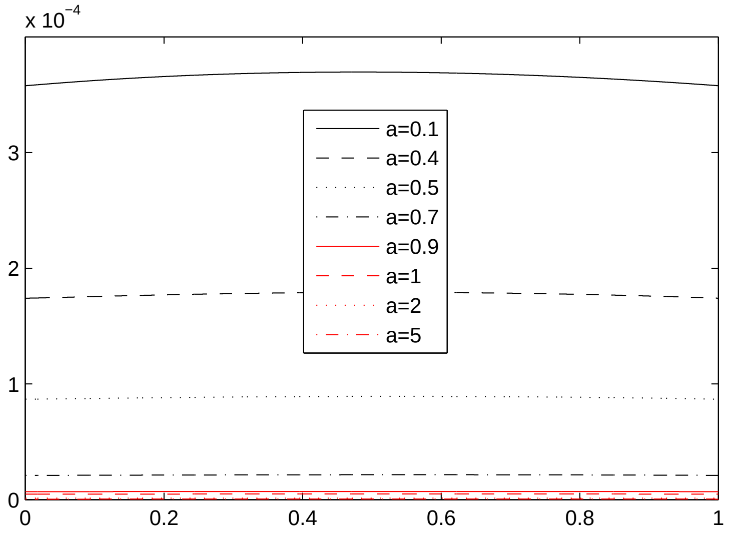


Figure 4.8: Example 2: Global errors for different values of a

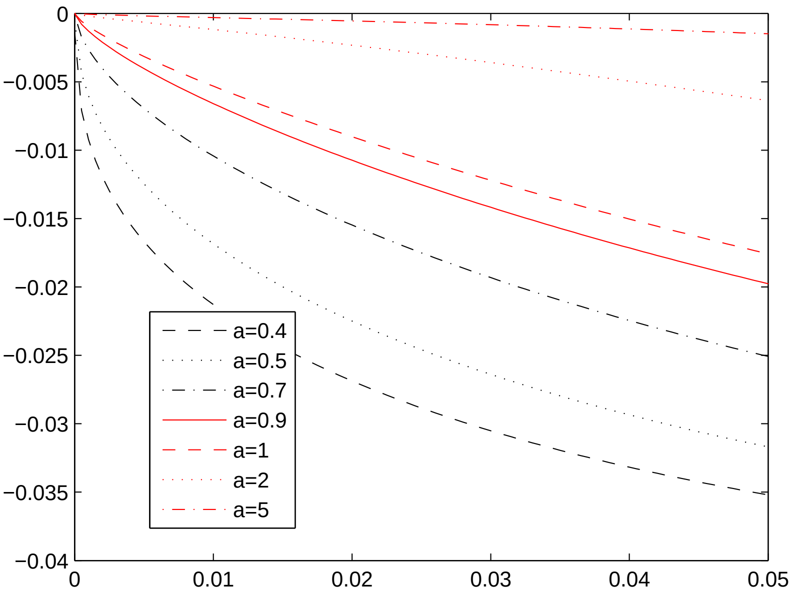


Figure 4.9: Example 2: Numerical values of the first derivative for different values of a

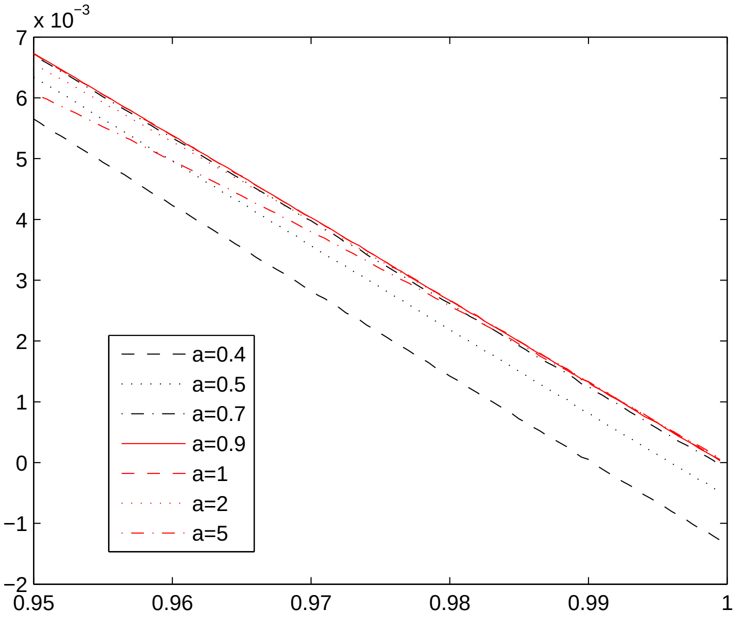


Figure 4.10: Example 2: Global errors for different values of a

4.2 Gas Permeation

Whether for biogas upgrading, natural gas upgrading or nitrogen production of air, gas permeation has become a major industrial application for membrane technology in the last 20 years. It is an important process in the modern chemistry and process engineering. Note that the results from this section are published in [5].

4.2.1 Theory

The theory of gas permeation can be found in [2]. In gas permeation, a gas mixture is passed across a membrane that is selectively permeable to one component of the incoming mixture, *feed*. Clearly, the *permeate* consists mainly of this very component. This process is illustrated in Figure 4.11.

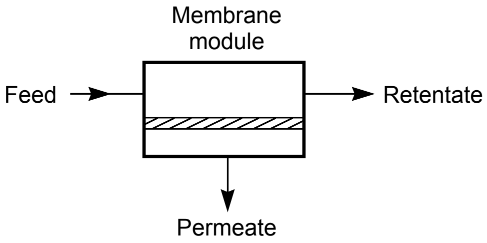


Figure 4.11: Schematic diagram of the membrane separation process, cf. [2], page 10

Membranes

We begin by quoting [2], page 3: In essence, a membrane is nothing more than a interface that moderates the permeation of a chemical species in contact with it.

For gas separation, either porous or dense membranes can be used. There are three types of porous membranes. Depending on the size of porse, gas permeates through this membranes by *convective flow*, *Knudsen diffusion* or *Molecular sieving*, see Figure 4.12.

In this master thesis, we deal with dense membranes where separation occurs by a *solution-diffusion mechanism*. Here the permeants dissolve in the membrane material (*solution*) and then diffuse through the membrane (*diffusion*) under the pressure's driving force. The permeants are separated because of the differences in the solubilities of the materials in the membrane and in the rates, at which the materials diffuse through the membrane. The ratio of the solubility and the diffusion rate is called the *permeability*.

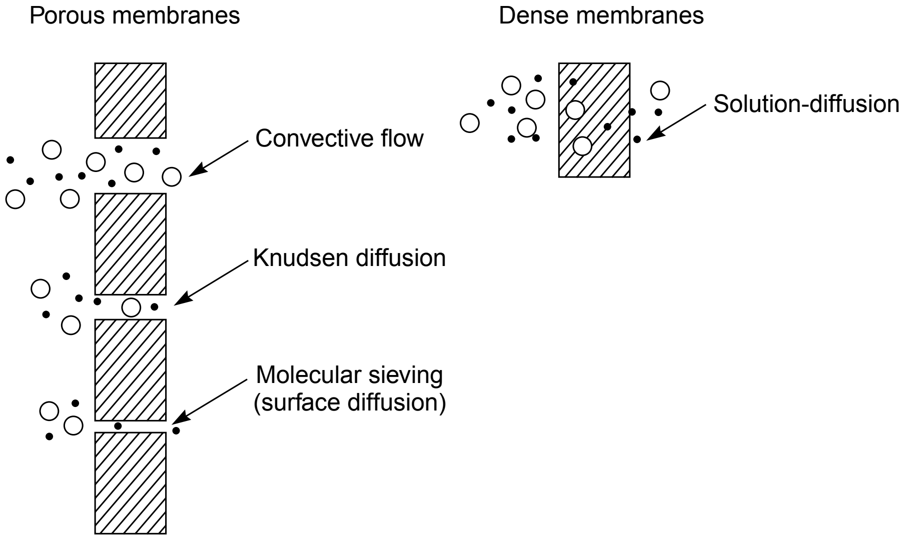


Figure 4.12: Permeation of gases through porous and dense membranes, see [2], page 303

Membranes and Modules

The membranes considered here are assumed to be hollow-fiber membranes. The diameters of fibers range from 50 to 3000 μm. Typically hollow-fiber membranes are packed into modules to enlarge the membrane surface, which is their major advantage.
There are two types of hollow-fiber membrane modules, cf. Figure 4.13. The first one is the *shell-side feed* module, where the fibers are arranged in a loop or a closed bundle. The feed reaches the bundle from the outside, then permeate passes through the fiber wall and exits through the open fiber ends. The second type is the *bore-side feed* module. In this case the fibers are open at both ends.

Configurations

Sometimes, a laterally flowing gas is used to change the composition of gas on the permeate side of the membrane. This gas sweeps the permeate off the membrane surface.

To avoid involving another gas, the gas mixture itself can be used as a sweep gas. There are two module configurations, *co-current* and *counter-current*, used in the process. If no sweep gas is used we speak of a *cross* configuration. An schematic diagram of all three configurations is given in Figure 4.14.

Multistage Systems

Often, for technical reasons, it is necessary to combine modules in a consecutive way. This means that the retentate is passed through several units.

4.2.2 Problem setting

In [14] following equations describing the gas permeation are formulated:

$$\frac{dF_i}{dt} = -Q_i, \quad \frac{dP_i}{dt} = Q_i, \tag{4.3}$$

where F_i and P_i are the volume flows of the component i in the feed channel and in the permeate channel respectively, t is the longitudinal coordinate of the membrane, and Q_i is the respective local trans-membrane flow,

$$Q_i = \Pi_i(x_i p_F - y_i p_P)s\pi d. \tag{4.4}$$

Here, s is the total number of fibres, d is the diameter of the active layer, Π_i is the permeance, p is the absolute pressure, and x_i and y_i are the volume fractions in the feed channel and in the permeate channel, respectively. The quantities x_i and y_i are given by

$$x_i = \frac{F_i}{\sum\limits_{i=1}^{k} F_i}, \quad y_i = \frac{P_i}{\sum\limits_{i=1}^{k} P_i}. \tag{4.5}$$

The differential equations (4.3) hold for a co-current configuration. For a counter-current configuration equations (4.3) take the form

$$\frac{dF_i}{dt} = -Q_i, \quad \frac{dP_i}{dl} = -Q_i.$$

Substituting (4.4), (4.5) into (4.3)yields

$$\frac{dF_i}{dt} = \Pi_i \pi s D \left(-\frac{F_i}{\sum\limits_{j=1}^{k} F_j} p_F + \frac{P_i}{\sum\limits_{j=1}^{k} P_j} p_P \right), \tag{4.6}$$

and

$$\frac{dP_i}{dt} = \Pi_i \pi s D \left(\frac{F_i}{\sum\limits_{j=1}^{k} F_j} p_F - \frac{P_i}{\sum\limits_{j=1}^{k} P_j} p_P \right). \tag{4.7}$$

In case of the counter-current configuration, we have to use the following equation instead of (4.7):

$$P_i' = \Pi_i \pi s D \left(-\frac{F_i}{\sum\limits_{j=1}^{k} F_j} p_F + \frac{P_i}{\sum\limits_{j=1}^{k} P_j} p_P \right). \tag{4.8}$$

We now rewrite the systems (4.6), (4.7) and (4.6), (4.8) in such a way that they match the notation used in previous sections, cf. (3.1a). Therefore, we set $z_{2i-1} := P_i$ and $z_{2i} := F_i$. In the solution vector for k gas components, $z = (z_1, ..., z_{2k})$, z_i with odd i represent the volume flows of single gas components in the feed channel and z_i with even i represent the volume flows of single gas components in the permeate channel.

Note that z_i is a function of t, $t \in [0, l]$, where l is the length of the module. To cover multistage systems with different module lengths, it is necessary to scale the interval $[0, l]$ to the normalized interval $[0, 1]$, which is done by multiplying the right hand side of the involved differential equations by l. The resulting differential equations for one gas component and co-current flow have the form

$$z_{2i-1}'(t) = \Pi_i \pi s D \left(-\frac{z_{2i-1}(t)}{\sum\limits_{j \text{ odd}} z_j(t)} p_F + \frac{z_{2i}(t)}{\sum\limits_{j \text{ even}} z_j(t)} p_P \right) l, \tag{4.9}$$

$$z_{2i}'(t) = \Pi_i \pi s D \left(\frac{z_{2i-1}(t)}{\sum\limits_{j \text{ odd}} z_j(t)} p_F - \frac{z_{2i}(t)}{\sum\limits_{j \text{ even}} z_j(t)} p_P \right) l. \tag{4.10}$$

For the counter-current flow the second equation has to be replaced by

$$z'_{2i}(t) = \Pi_i \pi s D \left(-\frac{z_{2i-1}(t)}{\sum\limits_{j \text{ odd}} z_j(t)} p_F + \frac{z_{2i}(t)}{\sum\limits_{j \text{ even}} z_j(t)} p_P \right) l. \qquad (4.11)$$

Now we formulate the necessary boundary conditions closing the system. At the gas inlet on the feed side of the membrane, the related boundary conditions read $z_{2i-1}(0) = \chi_i f$, where χ_i is the volume fraction of the i-th gas component in the gas mixture and f is the total gas volume flow. In the co-current case, if no sweep gas is introduced, the boundary conditions at the inlet to the permeate side of the membrane read $z_{2i}(0) = 0$. They are used with Equation (4.10). The initial conditions, $z_{2i-1}(0) = \chi_i f$, $z_{2i}(0) = 0$, can be written in the form of a linear system (3.1b), where B_0 is chosen as a $2k \times 2k$ identity matrix, B_1 as a $2k \times 2k$ zero matrix and $\beta = (\chi_1 f, 0, \chi_2 f, 0, \ldots, \chi_k f, 0)$.

In the case of the counter-current configuration, the boundary conditions for the feed side read $z_{2i-1}(0) = \chi_i f$ and for the permeate side they are $z_{2i}(1) = 0$. Now B_0 and B_1 are the following $2k \times 2k$ diagonal matrices: $B_0 = \mathrm{diag}(1, 0, 1, 0, \ldots, 1, 0)$ and $B_1 = \mathrm{diag}(0, 1, 0, 1, \ldots, 0, 1)$. The vector $\beta = (\chi_1 f, 0, \chi_2 f, 0, \ldots, \chi_k f, 0)$ remains unchanged.

4.2.3 Multistage Systems

In a multistage system, we apply Equations (4.9)-(4.11) separately for each stage. This means that for a S-stage system with k gas components, we have to solve $2Sk$ equations. Here, it is more difficult to adapt the boundary conditions. They depend on the structure of the system and have to be specified to reflect the module configuration.

For instance for a three component gas mixture permeating in the two-stage counter-current system shown in Figure 4.15, the boundary conditions are

$$\begin{aligned}
&z_1(0) = z_8(0) + \chi_1 f, \;\; z_2(1) = 0, \\
&z_3(0) = z_{10}(0) + \chi_2 f, \;\; z_4(1) = 0, \\
&z_5(0) = z_{12}(0) + \chi_3 f, \;\; z_6(1) = 0, \\
&z_7(0) = z_1(1), \;\; z_8(1) = 0, \\
&z_9(0) = z_3(1), \;\; z_{10}(1) = 0, \\
&z_{11}(0) = z_5(1), \;\; z_{12}(1) = 0.
\end{aligned}$$

4.2.4 Numerical Simulations

The results of the numerical simulations will be compared with experimental data taken from [14]. Let us first list the most important parameters. In the experiments, a hollow fiber bore-side module with dense membrane was used. It contains 800 polyamide fibers whose diameter is $0.0004\,m$, the length $0.38\,m$, which yields in a total membrane area of $0.38\,m^2$. Absolute pressure in feed was $9\,bar$, in permeate $1.1\,bar$. Both counter-current and co-current configuration without (external) sweep gas has been modeled. The feed gas composition and the permeances for the experiments are shown in Table 4.3.

Experiment		A	B	C	D
Feed gas composition [v/v]	CH_4	0.645	0.65	0.645	0.645
	CO_2	0.345	0.35	0.345	0.345
	O_2	0.01	–	–	0.01
	H_2O	–	–	0.01	–
Permeances [$m^3_{(stp)}$ / (m^2 s bar)]	CH_4	1.59e-6	1.59e-6	1.59e-6	1.59e-6
	CO_2	5.91e-5	5.91e-5	5.91e-5	5.91e-5
	O_2	1.36e-5	–	–	1.36e-5
	H_2O	–	–	3.2e-3	–
Membrane area [m^2]		0.38	0.38	0.38	0.38/0.75
Feed pressure [bar]		9.0	9.0	9.0	9.0
Permeate pressure [bar]		1.1	1.1	1.1	1.1
Feed flow [$L_{(stp)}$/min]		1 – 15	1	3	3.961
Flow configuration		both	counter	counter	counter

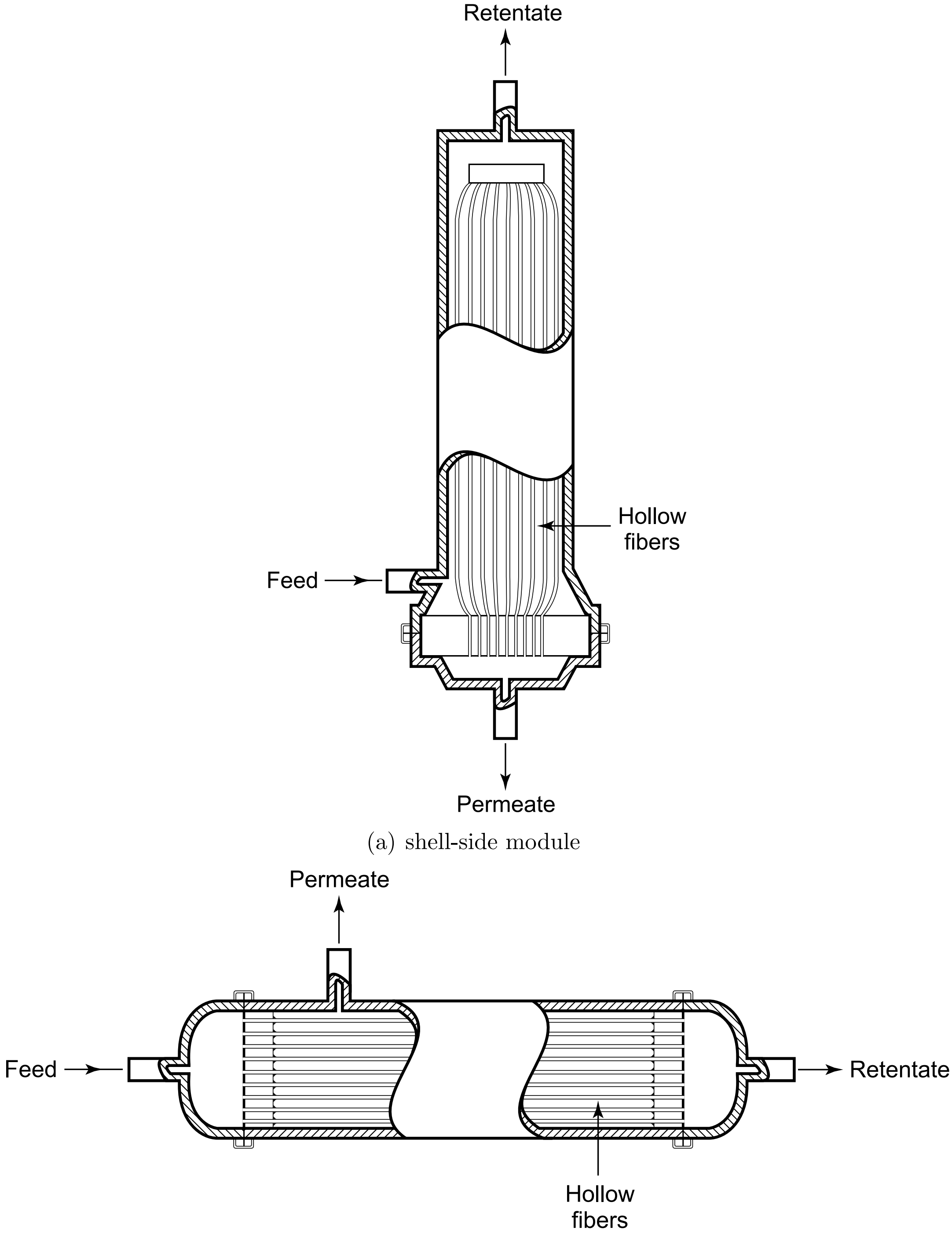
Table 4.3: Gas permeation parameters used in experiments and numerical simulations

Before discussing the numerical results, we have to introduce some characteristic quantities. The *purity of a gas i* in the gas mixture is defined as $\frac{x_i}{\sum x_i}$. The *recovery of a gas i* is the amount of gas i in the retentate divided by the amount of gas i in the feed. The *stage cut* is the ratio of the permeate volume flow to the feed volume flow.

Single-stage system with three gas components

In Figures 4.16 to 4.18, we compare numerical results obtained by means of the finite difference method presented in [14] with those calculated using the collocation code `bvpsuite`, see Section 3. The results show gas purity drawn against recovery or stage cut for both, co-current and counter-current flow configuration. The gas mixture consists of three gas components, methane, carbon dioxide, and oxygen. All significant process parameters for this study are presented in Table 4.3, column A. Figures 4.16 to 4.18

demonstrate that both algorithms, the experimentally verified finite difference method and the currently presented collocation method, provide virtually identical results. Slight differences originate from the different methods´ accuracies and the round-off errors. In Figures 4.19 and 4.20, the change in volume flow of each gas component over the module length can be seen.



(a) shell-side module

(b) bore-side module

Figure 4.13: Two types of hollow-fiber membrane modules used for gas separation, cf. [2], page 147

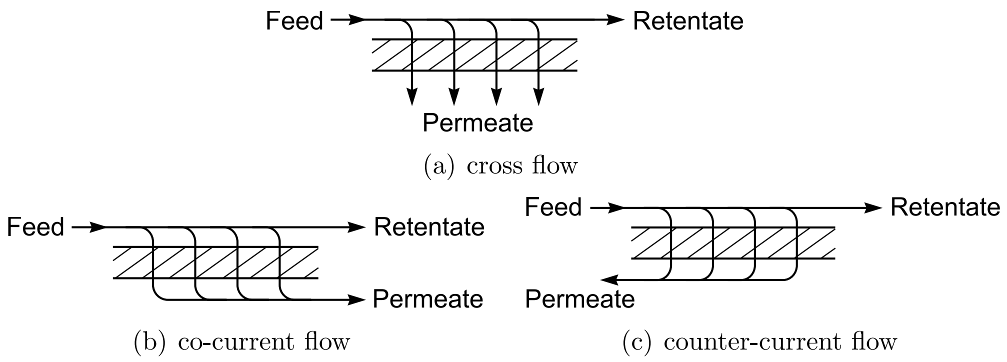


Figure 4.14: Configurations for gas permeation, see [2], page 185

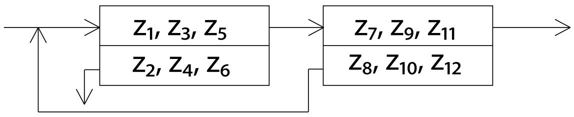


Figure 4.15: Two-stage counter-current system for three gas components

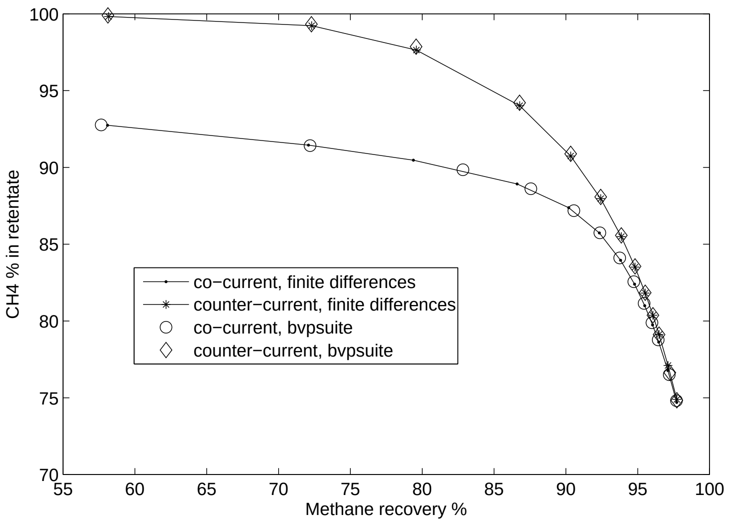


Figure 4.16: Methane concentration obtained from numerical simulation plotted versus methane recovery

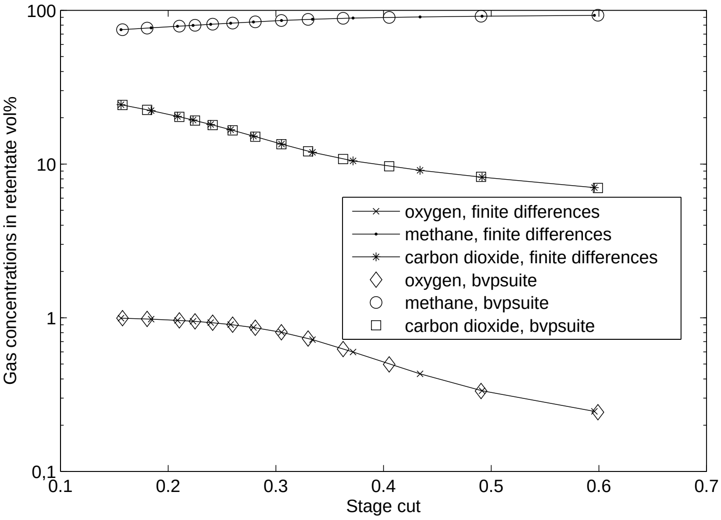


Figure 4.17: Gas concentration obtained from numerical simulation plotted versus stage cut for the co-current flow

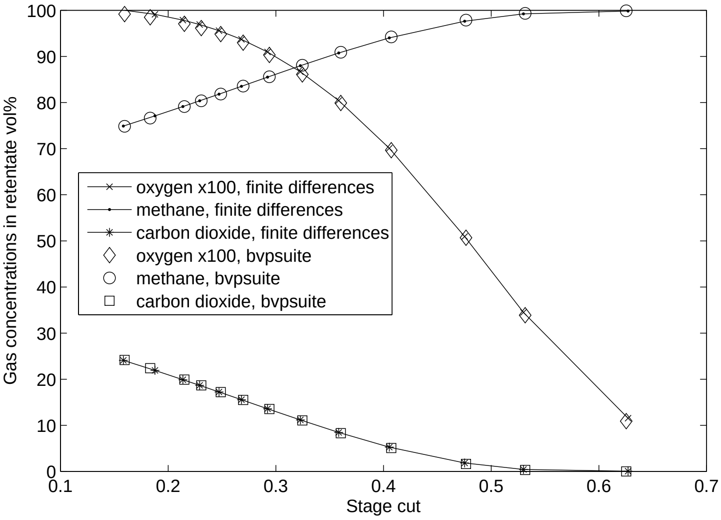


Figure 4.18: Gas concentration obtained from numerical simulation plotted versus stage cut for the counter-current flow

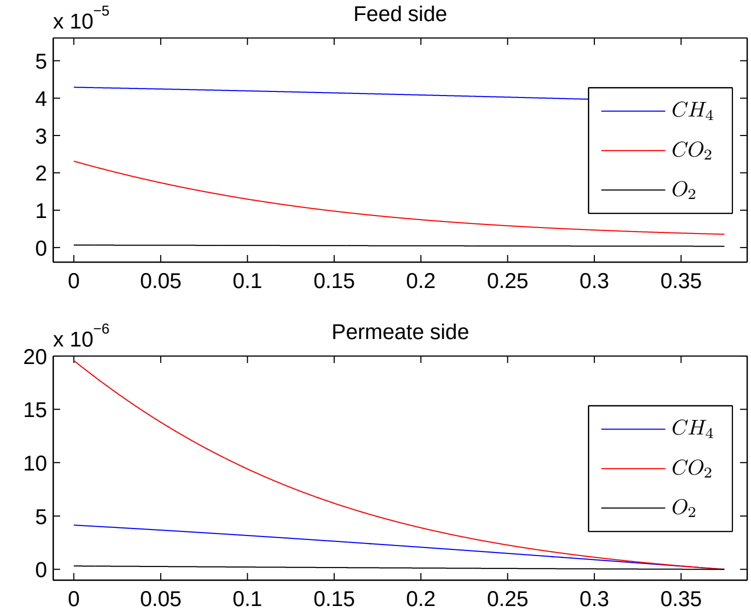


Figure 4.19: Results from `bvpsuite`: Change in volume flow over the module length for experiment A in counter-current configuration with a feed flow of 3.961

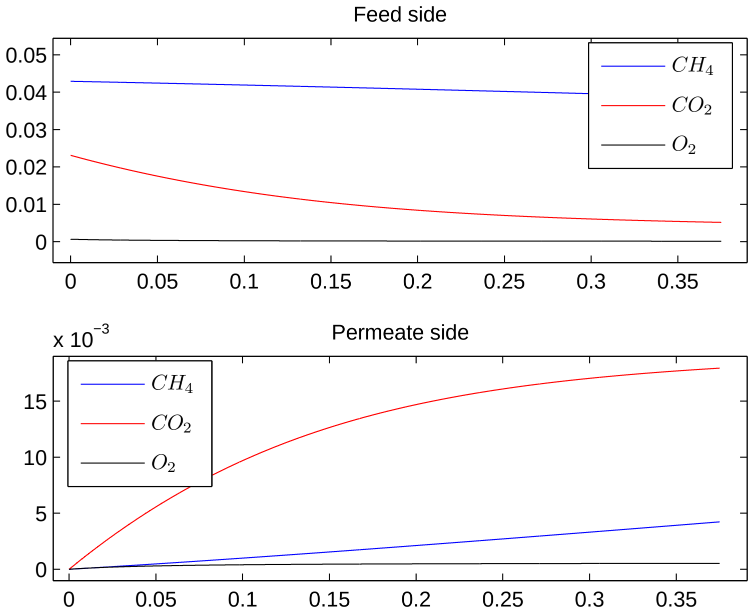


Figure 4.20: Results from `bvpsuite`: Change in volume flow over the module length for experiment A in co-current configuration with a feed flow of 3.961

Single-stage system with one fast permeating gas component

In this section, we deal with a process including a fast permeating gas component. The gas mixture in this experiment contains three gas components, methane, carbon dioxide, and water. The values of all relevant parameters are specified in detail in Table 4.3, column C. As we can see in Figures 4.21 and 4.22 the water permeates very rapidly through the membrane. This behavior requires a dense mesh in the area where the volume flow of the water changes rapidly. The adapted mesh produced by `bvpsuite` has this property, cf. Figure 4.22.

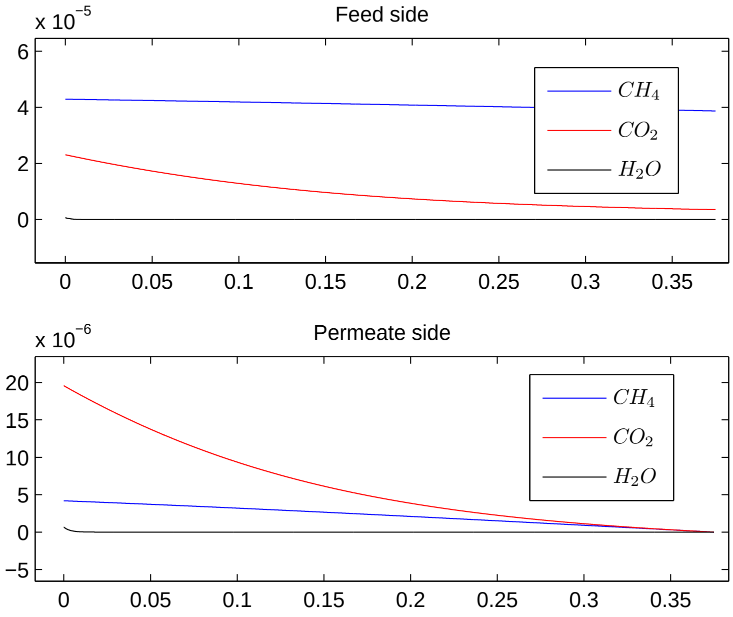


Figure 4.21: Results from `bvpsuite`: Change in volume flow over the module length for experiment C

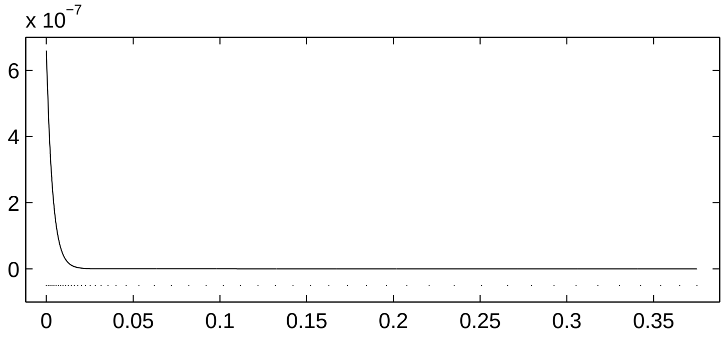


Figure 4.22: Results from `bvpsuite`: Change in volume flow and adapted mesh for the gas component H_2O, in detail, over the module length for experiment C

Two-stage system with variable lengths

In gas permeation, it is of particular interest to provide a retentate with high purity and high recovery rate. Therefore, we study a two-stage system (cf. Section 4.2.3), with fixed total length and a varying ratio between the module lengths (the length of the modules varies from 10 to 90 percent). The parameters are as in the experiment D.
In Figures 4.23 and 4.24, we compare the characteristic quantities purity and recovery, in case that the total length of both modules is $0.375\,m$ or $0.75\,m$, respectively. It can be easily seen that the quantities behave in a contrary way: when the purity grows the recovery rate drops and vice versa.

As a possible further research task, one could study the problem in a reformulated form, as an optimization problem.

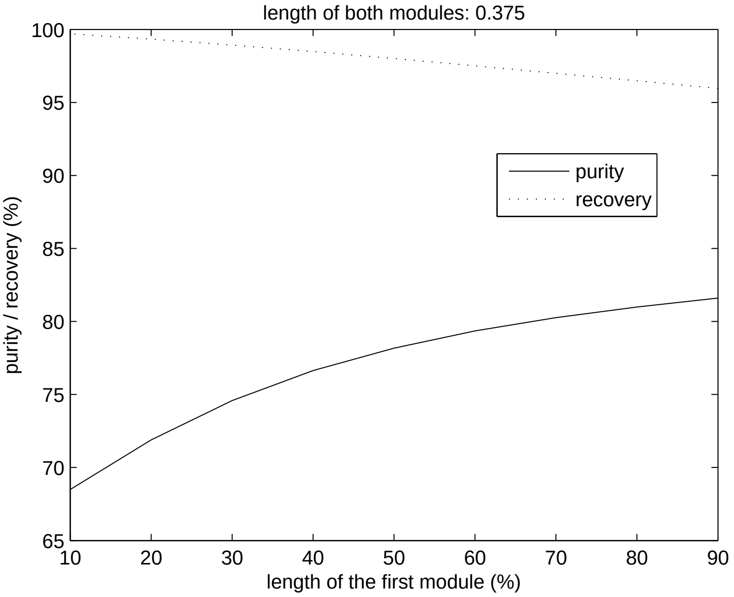


Figure 4.23: Results from `bvpsuite`: CH_4 purity and recovery plotted for a two-stage system varies in the length of the modules (total length of the two modules: 0.375)

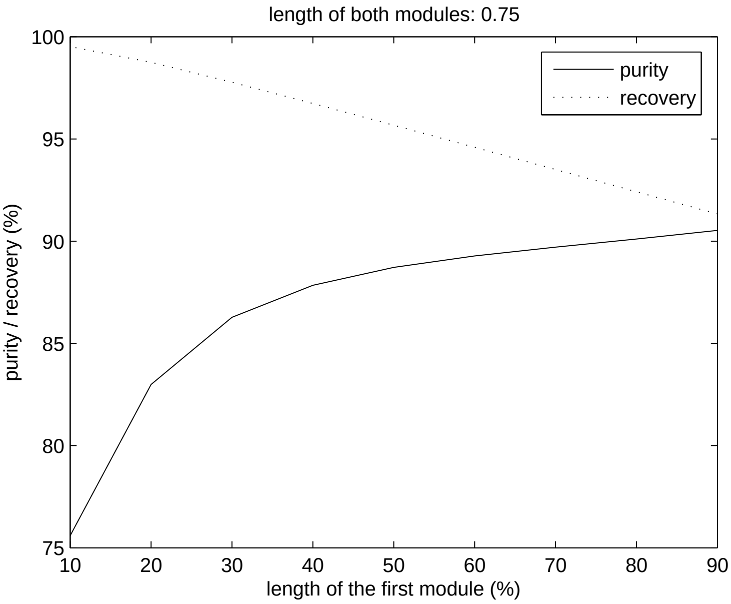


Figure 4.24: Results from `bvpsuite`: CH_4 purity and recovery plotted for a two-stage system varies in the length of the modules (total length of the two modules: 0.75)

Performance comparison

The advantage of `bvpsuite` is its high computational efficiency, especially in cases that involve pronouncedly different permeation rates. Let us first consider a permeation process of a gas mixture containing methane and carbon dioxide in counter-current configuration under the conditions specified in Table 4.3, column B. To reach the accuracy of 10^{-9} the finite difference underrelaxed method requires a grid with 1000 points and 24.7 seconds, while the collocation method provides the same result on a grid with 10 points and calculation time 2.2 seconds.

If a third gas component with a relatively high permeation rate is added (permeation study with H_2O defined in column C in Table 4.3), the performance is again strongly in favor of the collocation method. In this scenario, the collocation method solves the problem within around 6 seconds. The finite difference method requires several minutes to accomplish the same goal. The adaption of the computational grid in context of the collocation method is such that the solution behavior is reflected in a correct way. The grid becomes denser where the water partial pressure changes rapidly, around the feed inlet. This effect is captured very well by the grid adaptation strategy. The finite difference method struggles against the numerical diffusion that has a relatively strong effect in case of rapidly permeating water.

Bibliography

[1] W Auzinger, G. Kneisl, O. Koch, and E. Weinmüller. A Solution Routine for Singular Boundary Value Problems. Technical report, Institute for Applied Mathematics and Numerical Analysis, 2002.

[2] R.W. Baker. *Membrane Technology and Applications*. John Wiley & Sons, 2004.

[3] J. Cash, G. Kitzhofer, O. Koch, G. Moore, and E. Weinmüller. Numerical solution of singular two-point boundary value problems. *J. Numer. Anal. Indust. and Appl. Math.*, 4:129–149, 2009.

[4] C. de Boor and B. Swartz. Collocation at gaussian points. *SIAM J. Numer. Anal.*, 10:582–606, 1973.

[5] A. Feichtinger, A. Makaruk, W. Weinmüller, A. Friedl, and M. Harasek. Collocation method for the modeling of membrane gas permeation systems. *Int. J. Nonlinear Sci. Numer. Simul.*, 15(5):307–316, 2014.

[6] A. Feichtinger, I. Rachunková, S. Stanek, and E. Weinmüller. Periodic BVPs in ODEs with time singularities. *Comput. Math. Appl.*, 62(4):2058–2070, 2011.

[7] F.R. de Hoog and R. Weiss. Difference methods for boundary value problems with a singularity of the first kind. *SIAM J. Numer. Anal.*, 13:775–813, 1976.

[8] F.R. de Hoog and R. Weiss. Collocation methods for singular boundary value problems. *SIAM J. Numer. Anal.*, 15:198–217, 1978.

[9] G. Kitzhofer, O. Koch, and E. Weinmüller. Kollokationsverfahren für singuläre Randwertprobleme zweiter Ordnung in impliziter Form. Technical report, Institute for Applied Mathematics and Numerical Analysis, 2004.

[10] G. Kitzhofer, O. Koch, and E. Weinmüller. Numerical Treatment of Singular BVPs: The New MATLAB Code bvpsuite. *AIP Conference Proceedings*, 1168(1):39–42, 2009.

[11] G. Kitzhofer, O. Koch, and E. Weinmüller. Pathfollowing for essentially singular boundary value problems with application to the complex Ginzburg–Landau equation. *BIT Numerical Mathematics*, 49:141, 2009.

[12] O. Koch. Asymptotically correct error estimation for collocation methods applied to singular boundary value problems. *Numer. Math.*, 101:143–164, 2005.

[13] O. Koch, R. März, D. Praetorius, and E.B. Weinmüller. Collocation methods for index-1 DAEs with a singularity of the first kind. *Math. Comp.*, 79(269):281–304, 2010.

[14] A. Makaruk and M. Harasek. Numerical algorithm for modelling multicomponent multipermeator systems. *Journal of Membrane Science*, 344(1-2):258–265, 2009.

[15] G. Pulverer, G. Söderlind, and E. Weinmüller. Automatic grid control in adaptive BVP solvers. *Numer. Algorithms*, 56:61–92, 2011.

[16] M. Schöbinger and S. Schirnhofer. Singuläre skalare DG 1. Ordnung, April 2012.

[17] E. Weinmüller. Analysis of Singular BVPs in ODEs. Talk at Palacky University, Olomouc, Czech Republic, 25-27.11.2013.