MICHAEL WULKOW

Numerical Treatment of Countable Systems
of Ordinary Differential Equations

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Str. 10
D-10711 Berlin-Wilmersdorf
Telefon: 030-89604-0
Telefax: 030-89604-125
e-mail: bibliothek@sc.zib-berlin.de

Verantwortlich: Dr. Klaus André

ISSN 0933-789X
Numerical Treatment of Countable Systems of Ordinary Differential Equations

MICHAEL WULKOW

Konrad-Zuse-Zentrum für Informationstechnik Berlin, Heilbronner Strasse 10, D-1000 Berlin 31, Federal Republic of Germany

December 1990

ABSTRACT

Countable systems of ordinary differential equations appear frequently in chemistry, physics, biology and medicine. They can be considered as ordinary differential equations in sequence spaces. In this work, a fully adaptive algorithm for the computational treatment of such systems is developed. The method combines time discretization with extrapolation in Hilbert spaces with a discrete Galerkin approach as discretization of the stationary subproblems. The Galerkin method is based on orthogonal functions of a discrete variable, which are generated by certain weight functions. A theory of countable systems in the associated weighted sequence spaces is developed as well as a theory of the Galerkin method. The Galerkin equations can be assembled either by use of analytical properties of the orthogonal functions or numerically by a multilevel summation algorithm. The resulting algorithm CODEX is applied to many examples of technological interest, in particular from polymer chemistry.
## CONTENTS

**Introduction** 1

1 **Countable Systems of Ordinary Differential Equations** 4
   1.1 Introductory Considerations 4
   1.2 A Two-Parameter Scale of Hilbert Spaces 13
   1.3 Theory of Countable Systems 18
   1.4 Discretization and Extrapolation 24

2 **Modified Discrete Laguerre Polynomials** 29
   2.1 Construction of the Polynomials 30
   2.2 Basic Properties of the Polynomials 35
   2.3 Approximation Properties 39

3 **Discrete Galerkin Method** 43
   3.1 Projections and Approximate Solutions 43
   3.2 Weight Function Fitting 49

4 **Numerical Realization of the Algorithm** 53
   4.1 Realization of the Time-Step Control 53
   4.2 Realization of the Galerkin Method 60
   4.3 The Program CODEX 62

5 **Numerical Examples** 64

6 **Appendix: A Multilevel Summation Algorithm** 81

References 88
INTRODUCTION

An initial value problem for a countable system of ordinary differential equations (CODE) consists of a (in general infinite) sequence of single ordinary differential equations. Hence a CODE can be considered as an ordinary differential equation in a sequence space. The numerical treatment of such systems has become more and more interesting, since CODE's appear frequently in chemistry, physics, biology and medicine. For example, the modeling of a polymerization process in chemistry, where single molecules (monomers) are linked together to long polymer chains (consisting of up to $10^6$ monomers), leads to as many single equations as polymer chains can arise during the reaction. Since in general this number is not known a priori (sometimes not even the scale), it often must be assumed to be infinite. Similar models can be found for soot formation mechanisms arising in smog reactions [21] or combustion [31] and for dust formation in stellar winds [22] in astrophysics. Other examples are master-equations in biology and chemistry [28], branching processes [12] and Kolmogorov equations [29] in statistics. Common to all these problems is, that the solutions (as far as they are available) have a special structure, e.g. they can be interpreted as probability densities which are in some sense similar to well-known probability distributions. This has been exploited to get quantitative results for special applications. However, the methods in use are frequently only valid for a small class of problems, because a lot of additional assumptions are necessary. For example the method of moments [21] applied for problems of polyreaction kinetics needs the type of the distribution expected to be a solution, the so-called lumping [4] requires deep insight into the process and introduces unknown modeling errors in the case of nonlinear problems. The popular quasi-steady-state approximation (QSSA), which can be regarded as a change of the modeling has disadvantages discussed elsewhere [19], [9]. The use of a direct large scale integration usually is impossible for storage and time reasons. Other techniques are the so-called continuous modeling (transforming the CODE into a partial differential equation), which leads to theoretical difficulties [23] and the use of discrete Fourier transforms, which are very time-consuming [39].

Mathematical theory concerning CODE's has been developed for many years, a survey is given in the monograph of Deimling [12]. Nevertheless, the computational treatment of CODE's has not become as important as the treatment of partial differential equations (PDE's) so far – a fact that can be recognized by looking into textbooks about abstract differential equations, which actually should cover both PDE's and CODE's. Moreover, there are no numerical methods which try to be a general approach to this class of problems and 'nearly nothing has been done concerning qualitative behavior of solutions' [12].
As a really new approach a *discrete Galerkin method* has been suggested by DEUFLHARD and WULKOW [18]. This Galerkin method is based on orthogonal polynomials of a discrete variable constructed by means of a weighted discrete inner product. The discrete variable represents the sequence index of the CODE and plays the role of a ‘space’ variable in a partial differential equation. The weight function determining the inner product is chosen to reflect certain structures of the problem. Besides it depends on a time-depending parameter, which can be adapted by a so-called *moving weight function* condition. As examples for such weight functions the geometric distribution and the Poisson distribution have been used in [18]. In this way, the countable system is discretized by the Galerkin method in a weighted sequence space, such that differential equations for the Galerkin coefficients are obtained by use of analytical properties of the orthogonal polynomials. The discrete Galerkin method is comparable to a *method of lines* with *moving basis functions* in the field of partial differential equations. In [9] and [52] the method has been developed to such a level, that problems of technological interest – polymerization of Methyl Methacrylate (modeled by a system of CODE’s), heterogeneous polymer degradation – can be solved very efficiently now. Additionally, a computational cheap a posteriori estimation of the approximation error can be obtained.

In the approach sketched above there are some open points, concerning the theory of CODE’s in certain weighted sequence spaces, the convergence theory of the Galerkin method in these spaces, the adaptive control of the Galerkin approximations and the treatment of problems containing different types of weight functions. The present work clarifies these questions and extends the Galerkin method to more general problems. This is done by combining concepts from the numerical treatment of ODE’s with adaptive approaches from PDE’s. An important aspect is the fact, that a CODE resembles more to a PDE with a discrete space variable than to an extension of a finite dimensional ODE system. Discretization first in time, in theory of PDE’s better known as Rothe’s method [49], leads to adaptive time-steps really belonging to the sequence space problem – following ideas of Bornemann [8]. This technique permits an adaptive matching between the requirements of the time-step control and the ‘space’ discretization. As time discretizations explicit and (semi-) implicit Euler schemes are used, the occurring subproblems (projection into the sequence spaces, countable systems of algebraic equations) are attacked by a new (i.e. extended) discrete Galerkin method. For that, a two-parameter family of weight functions is introduced, connecting properties of some basic probability distributions. These weight functions lead to a two-parameter scale of weighted sequence spaces, for which the theory of CODE’s is developed.
Finally an algorithm is derived, which allows even the treatment of problems without knowledge of analytical properties.

It turns out, that the algorithm works well for a wide class of problems with solutions having structural similarities to the family of weight functions.

In Chapter 1 some general results from the theory concerning CODE's are collected. Some opening considerations introduce operators arising in typical examples and motivate the choice of certain sequence spaces $H_{p,a}$. These sequence spaces will be the key to the connection between the theory about existence and uniqueness of solutions of CODE's and the numerical approximation of solutions. Asymptotic expansions of Euler discretizations in $H_{p,a}$ are given next. Chapter 2 contains the construction of the basis functions of $H_{p,a}$, which are determined by the so-called modified discrete Laguerre polynomials. Important properties of these polynomials are included. Discrete Galerkin methods for projections in $H_{p,a}$ and approximate solutions of countable systems of algebraic equations (CAE's), both arising from Euler discretizations of CODE's, are described in Chapter 3. The numerical realization of the algorithm is documented in Chapter 4 and Chapter 5 illustrates on numerical results the efficiency of the algorithm developed herein. As an appendix the concept of the multi-level summation algorithm SUMMATOR, which is the heart of the numerical pre-processing suggested in Chapter 3, is explained in Chapter 6.

Acknowledgements. In the first place I want to thank Prof. Dr. P. Deuflhard. He lead me to this subject and supported my research all the time. Many of the ideas and concepts he taught me in the last years have entered in this work. I thank Folkmar Bornemann for numerous discussions, hints, ideas and suggestions concerning the subject and the manuscript. My program could be implemented very fast on the basis of his code KASTIX. Many thanks to Dr. Jörg Ackermann. The discussions with him gave me a lot of insight, in particular in view of the numerical examples. Many discussions with and encouragement by Dr. Ralf Kornhuber were very helpful for me. The interaction of the persons in the several groups of the Konrad-Zuse Zentrum has been very important for this work. In particular I want to mention all members of the numeric group and the symbolic group, without whose version of REDUCE a lot of the results in Chapter 2 could not (or only with difficulties) have been found out. Last but not least I want to thank Regina Telgmann for all her encouragement, listening and her repeated reading of the manuscript.
1 Countable Systems of Ordinary Differential Equations

Following the notation of DEIMLING [12] essentially, this work will be concerned with scalar initial value problems of the type (CODE)

\[(1.1) \quad u'_s(t) = f_s(t, u_1(t), u_2(t), \ldots), \quad u_s(0) = \varphi_s, \quad s \in \mathbb{N} \quad (s \geq 1),\]

where the functions

\[f_s : [0, T] \times D \rightarrow \mathbb{R}, \quad D \subseteq \mathbb{R}^\mathbb{N}, \quad s \in \mathbb{N},\]

and the initial value

\[\varphi = (\varphi_s) \in \mathbb{R}^\mathbb{N},\]

are given. The actual sequence space will be specified later on. The prime denotes the derivative with respect to the time \(t\). The ‘space’ variable, i.e. the index of the sequence, will usually be called \(s\) in this work, in imitation of the notations used in applications. For ease of writing we will alternate between the notations \(u_s(t)\) and \(u(s, t)\) for the \(s\)-component of the sequence (grid function) \(u\) at time \(t\). As far as the context is clear, the time-dependence will be omitted, such that \(u(s)\) or \(u_s\) means \(u(s, t)\).

A function

\[u : [0, T] \rightarrow D, \quad T \in (0, T],\]

is called a solution of (1.1), if \(u_s(0) = \varphi_s, \quad u_s \in C^1((0, T])\) and \(u'_s = f_s(t, u)\) in \([0, T]\) for each \(s \in \mathbb{N}\). In connection with time discretization and extrapolation in Hilbert spaces, we will mainly feature the linear case

\[(1.2) \quad u'(t) = A(t) u(t) + h(t), \quad u(0) = \varphi, \quad t \in (0, T],\]

with \(A\) a linear operator on certain sequence spaces and \(h\) a sequence-valued \(L^1\) - function. Problem (1.2) is a (linear) evolution problem. The existence and uniqueness theory will be given for the general nonlinear case.

Before we will be engaged in the theory of such problems, we will discuss some examples first, which will draw our attention to typical difficulties occurring with CODE’s, showing that CODE’s are not just ODE’s.

1.1 Introductory Considerations

This section serves as a motivation of the theory developed in the following parts of Chapter 1. Thus some details of the presentation will be specified in later sections. We study first typical examples of countable systems of ordinary differential equations.
Example 1.1: Backward difference equation. Consider the equation

\[ u'(t) = -\nabla u(t) , \quad u(0) = \varphi , \]

given in some sequence space, where \( \nabla \) defined by

\[ (\nabla u)_s = u_s - u_{s-1} , \quad (\nabla u)_0 = u_1 , \]

denotes the backward difference operator. Equation (1.3) appears as a basic module in many problems. For an initial sequence \( \varphi \) the solution of (1.3) can be written as

\[ u_s(t) = (T(t) \varphi)(s) \]

in terms of a semigroup (compare Section 1.4) \( T(t) \) given by

\[ (T(t) \varphi)(s) = e^{-t} \sum_{r=1}^{\infty} \frac{t^{s-r}}{(s-r)!} \varphi(r) . \]

Specializing \( \varphi_s = \delta_{s,1} , \quad \delta_{s,r} \) the Kronecker symbol, for fixed \( t \) the solution \( u(t) \) reduces to a Poisson distribution with parameter \( t \):

\[ u_s(t) = e^{-t} \frac{t^{s-1}}{(s-1)!} . \]

Thus in general \( u(t) \) can be considered as a convolution of Poisson distributions. In [18] the problem (1.3) has been treated as a chain addition polymerization in a space essentially spanned by Charlier polynomials. These polynomials are orthogonal with respect to a scalar product induced by the Poisson distribution.

Example 1.2: Summatory systems. Equations of this type have been studied by Hille [30] and can be regarded as special settings of mathematical models of polymer degradation processes [4], which have been solved numerically by use of the discrete Galerkin method in [18] and [52]. Let us consider the following special summatory system

\[ u'_{s}(t) = -(s-1)u_{s}(t) + \sum_{r=s+1}^{\infty} u_{r}(t) , \quad u_{s}(0) = \varphi_{s} , \quad s \in \mathbb{N} . \]

Suppose that for a given initial value \( \varphi \in l^1 \) the equation (1.7) has at least one solution \( u(t) \in C^1([0,T],l^1) \), defined for positive values of \( t \) such that \( u_{s}(t) \to \varphi_{s} \) for \( t \to 0 \). We define

\[ u_{s}(t) := \sum_{r=s}^{\infty} u_{r}(t) . \]
The series on the right-hand side converges uniformly, since \( u(t) \in C^1((0,T], l^1) \) by assumption and we can insert \( u_s(t) = v_s(t) - v_{s+1}(t) \) into the \( s \)-th equation of the system (1.7). This yields the relation

\[
(1.8) \quad v'_s(t) + (s - 1) v_s(t) = v'_{s+1}(t) + s v_{s+1}(t),
\]

which must be independent of \( s \). If we denote the common value of both sides of (1.8) by \( f(t) \), we can compute \( v_s(t) \) by solving the linear ordinary differential equation

\[
v'_s(t) + (s - 1) v_s(t) = f(t),
\]

which easily leads to

\[
v_s(t) = v_s(0) e^{-(s-1)t} + \int_0^t e^{(s-1)(\tau-t)} f(\tau) \, d\tau, \quad s \in \mathbb{N}.
\]

Finishing up with the re-substitution of the \( u_s \) we get

\[
(1.9) \quad u_s(t) = v_s(0) e^{-(s-1)t} - v_{s+1}(0) e^{-st} - \int_0^t e^{s(\tau-t)} [1 - e^{t-\tau}] f(\tau) \, d\tau.
\]

This shows, that the initial value problem (1.7) has solutions depending on an arbitrary function \( f \), which only has to be integrable. From [30] we know, that (1.9) is in fact a solution in \( l^1 \). The task now is to find a sequence space, in which the solution is unique. For this, we define a family of Hilbert spaces

\[
(1.10) \quad H_t := \left\{ u \in \mathbb{R}^\mathbb{N} \mid \|u\|^2_t := \sum_{s=1}^{\infty} u_s^2 e^{st} < \infty \right\},
\]

and show, that the condition \( u(t) \in H_t \) enforces uniqueness of the solution. A short calculation yields, that

\[
\sum_{s=1}^{\infty} \left( v_s(0) e^{-(s-1)t} - v_{s+1}(0) e^{-st} \right)^2 e^{st}
\]

is bounded for \( \varphi \in l^1, \ t \in [0,T] \). Thus we can show that \( u(t) \in H_t \), if

\[
(1.11) \quad \sum_{s=1}^{\infty} \left( \int_0^t e^{s(\tau-t)} [1 - e^{t-\tau}] f(\tau) \, d\tau \right)^2 e^{st}
\]

is bounded. This implies that

\[
\int_0^t e^{s(t-\frac{t}{s})} [1 - e^{t-\tau}] f(\tau) \, d\tau \to 0 \quad \text{for} \ s \to \infty,
\]
leading in turn to the condition

\[ (1.12) \quad \left| \int_0^t e^{\tau} g(\tau) d\tau \right| \leq M , \quad s = 1, 2, \ldots, \]

for a constant \( M \) and \( g(\tau) := (1 - e^{t-\tau}) f(\tau) \). Extension of the proof of Lemma 4.1.1 in [44] to integrable functions allows to conclude from (1.12), that \( g(\tau) \equiv 0 \) on \([0, t]\) almost everywhere. This means, that by restriction from \( l^1 \) to the space \( H_t \) we obtain a unique solution

\[ u_s(t) = v_s(0) e^{-\tau} - v_{s+1}(0) e^{-st}. \]

If we compare this result with the numerical solution of degradation processes achieved by the discrete Galerkin method in [18], we observe that there a so-called moving weight function condition in principle has led to the space \( H_t \). As we will see below (Corollary 1.5), the condition \( u \in H_t \) in particular implies, that all statistical moments of \( u \) are bounded. This is a natural assumption for many problems. In addition, we have seen in [18], that in \( H_t \) an efficient approximation of solutions of (1.7) is possible. So for theoretical and for numerical reasons the choice of the spaces \( H_t \) seems to be promising.

In the above example we got uniqueness by restricting the solution to ‘smaller’ spaces. We will exemplify now, that in order to obtain existence of solutions it can be necessary to extend to ‘larger’ spaces (see also examples in [12]):

**Example 1.3: Discrete convolution.** Consider the problem

\[ (1.13) \quad u'_s(t) = \sum_{r=1}^{s-1} u_r(t) u_{s-r}(t) , \quad u_s(0) = \varphi_s , \]

defined by means of the convolution operator \( A_C \)

\[ (1.14) \quad (A_C u)_s := \sum_{r=1}^{s-1} u_r u_{s-r} . \]

This operator is nonlinear and plays a role in many applications, where objects (e.g. chemical molecules) of size \( r \) and size \( s - r \) are combined to an object of size \( s \) (combination or coagulation processes).

Let us assume that the initial value

\[ (1.15) \quad \varphi(s) := (1 - \rho) \rho^{s-1} , \quad 0 < \rho < 1 , \]

is prescribed. Such an initial setting is typical for the problems treated in Chapter 5 and convenient for our considerations anyway. We can check, that

\[ \varphi \in H_\rho := \left\{ u \in \mathbb{R}^\mathbb{N} \mid \|u\|^2 := \sum_{s=1}^{\infty} u_s^2 (1 - \rho)^{-1} \rho^{-(s-1)} < \infty \right\} . \]
The family of sequence spaces $H_p$ is an extension of the family $H_t$ in Example 1.2. It is easily seen, that the solution of (1.13) is given by

$$u_s(t) = (t(1 - \rho) + \rho)^{s-1} (1 - \rho).$$

We can also show, that $u(t) \in H_p$ only for $(t(1 - \rho) + \rho)^2 < \bar{\rho} < 1$ and $t < 1$. So for the initial value (1.15) we only have a solution of (1.13) in $H_p$ if

$$t < \frac{\sqrt{\bar{\rho} - \rho}}{1 - \rho}.$$

A numerical consequence of this effect has been observed in [18], p. 296, where a CODE with an extended convolution operator is treated. There it is pointed out, that on the one hand an efficient Galerkin method for such problems can be realized in the spaces $H_p$, on the other hand the numerical approximations are breaking down for fixed $\bar{\rho}$ and increasing $t$. We will explain in Section 1.3, that the operator $A_C$ is not Lipschitz continuous as an operator on $H_p$ for fixed $\rho$, but as an operator on the scale $H_p$, $0 < \rho < 1$.

**Computational approach: Discrete Galerkin method.** The preceding examples give a first impression of the characteristics of CODE’s. Now we turn to the numerical treatment of such problems and give a brief summary of the discrete Galerkin method introduced in [18]. The details will become evident in Chapter 2 and Chapter 3.

We assume for the solution $u(t)$ of a CODE, that (compare Example 1.3)

$$u(t) \in H_p := \left\{ u \in \mathbb{R}^N \mid \|u\|^2 = \sum_{s=1}^{\infty} u_s^2\Psi_p(s)^{-1} < \infty \right\}$$

where $\Psi_p(s)$ is a positive weight function with a (possibly time-dependent) real parameter $\rho$. On certain conditions on the weight function $\Psi_p$, $H_p$ is a separable Hilbert space equipped with the scalar product

$$\langle u, v \rangle_p := \sum_{s=1}^{\infty} u(s)v(s)\Psi_p^{-1}(s).$$

The orthogonal basis $\{\psi_j(\rho)\}_j$ of $H_p$ can be written as

$$\psi_j(s; \rho) = \Psi_p(s) l_j(s; \rho), \ j = 0, 1, \ldots, s \in \mathbb{N},$$

with the $\{l_j(s; \rho)\}_{j=0,1,\ldots}$ a set of polynomials of a discrete variable $s$.

These polynomials are associated with $\Psi_p(s)$ by the orthogonality relation

$$\langle \psi_j(\rho), \psi_k(\rho) \rangle_\rho = \sum_{s=1}^{\infty} l_j(s; \rho) \Psi_p(s) = \gamma^\rho_j \delta_{jk}, \gamma^\rho_j > 0,$$
just the countable system (1.3). In the following we will denote the concentration of a polymer of degree \( s \) at time \( t \) by \( P_s(t) \) respectively \( D_s(t) \).

The modeling of free radical polymerization processes leads to a system of CODE's, with one component \( P_s(t) \) denoting the so-called radical polymer and the second component \( D_s(t) \) describing the dead polymer. The complete reaction consists of some single steps, e.g. chain addition, combination (convolution), chain transfer and an additional source term. A model of technological relevance (MMA polymerization) has to be described by a system of at least \( 2 \cdot 10^4 \) ordinary differential equations. By means of the discrete Galerkin method this number can be reduced to \( 10 - 15 \) differential equations for a solution with technical accuracy \( (10^{-2}) \) and controlled approximation error. This makes the simulation of the complete model possible even on a small workstation. In this case, the Galerkin approximations are based on the geometric distribution, which is the distribution of choice for free radical polymerization processes. As shown in [9] the error estimation works well for the dead polymer and most of the process time for the radical polymer too. But there remains one gap, which is one of the starting points of this work. Table 1.1 shows the error estimation \( \varepsilon_n \) of the Galerkin approximation of the radical polymer in the first 5 seconds of the process for truncation indices \( n = 5 \) and \( n = 10 \). The dispersion coefficient \( d \) is a measure for the width of the distribution \( P_s(t) \) (see Chapter 2).

<table>
<thead>
<tr>
<th>time ( t )</th>
<th>( \varepsilon_5(t) )</th>
<th>( \varepsilon_{10}(t) )</th>
<th>( d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^{-2} )</td>
<td>( 3 \cdot 10^{-2} )</td>
<td>( 7 \cdot 10^{-2} )</td>
<td>1.35</td>
</tr>
<tr>
<td>( 10^{-1} )</td>
<td>( 3 \cdot 10^{-2} )</td>
<td>( 9 \cdot 10^{-2} )</td>
<td>1.34</td>
</tr>
<tr>
<td>( 5 \cdot 10^{-1} )</td>
<td>( 2 \cdot 10^{-2} )</td>
<td>( 3 \cdot 10^{-2} )</td>
<td>1.43</td>
</tr>
<tr>
<td>( 1 )</td>
<td>( 8 \cdot 10^{-2} )</td>
<td>( 5 \cdot 10^{-2} )</td>
<td>1.61</td>
</tr>
<tr>
<td>( 5 )</td>
<td>( 0.06 \cdot 10^{-2} )</td>
<td>( 0.01 \cdot 10^{-2} )</td>
<td>2.00</td>
</tr>
</tbody>
</table>

Table 1.1 Error estimations at the beginning of a polymerization

We observe that the error estimation does not look reasonable between \( t = 0.01 \) and \( t = 1.0 \). As an explanation, we note that depending on the values of the respective rate coefficient and the source term, the chain addition, the so-called transfer and termination terms or a mixture of all these effects may dominate the process during the initial phase. This can lead to solutions similar to the Poisson distribution (compare Example 1.1), the geometric distribution (see Example 1.3) or even to a piecewise nearly constant solution. Thus it can happen, that a Galerkin method based on the geometric distribution yields bad results. Only for \( t > 1 \) the termination steps of a free radical system definitely play the dominating role, increasing the concentration of short polymers. The chain length distribution \( P_s(t) \) becomes similar to a geometric distribution after
a while. Even though in some cases $P_s(t)$ cannot be represented well by a series (1.22) for a (short) time with $\Psi$ the geometric distribution, it turns out that the information contained in the expansion coefficients is sufficient (or not necessary) to ensure good approximations for $t \gg 1$.

As a consequence we make the demand:

- Find weight functions $\Psi$ which allow the treatment of processes containing different types of basic probability distributions.

Further we note, that there are models in polymer chemistry as well as in smog simulation or in statistics with reaction coefficients (transfer probabilities) depending on the present sequence index $s$ itself. It seems very difficult (or impossible) to find a weight function and associated polynomials with appropriate properties for a sufficient general case. For special heterogeneous reactions an attempt has been made in [52], but there analytical properties of the orthogonal polynomials have been used again. Therefore our second requirement is:

- Give an algorithm which allows a numerical preprocessing of a CODE.

However, there are at least two things which seem to prevent such a numerical preprocessing: First, the numerical evaluation of scalar products as needed in (1.24) requires the time consuming summation over a very large or infinite domain. Secondly, even if we can compute a numerical approximation of the right-hand side of the ODE for the expansion coefficients, a sophisticated ODE solver will become inefficient or will fail, since it will detect a defect of smoothness. Besides that, a method of lines has the disadvantage of not looking at the problem in the ‘right’ space, but only in an $\mathbb{R}^n$-setting (even if an Hilbert space error-norm and special scalings are used).

The above considerations have entered the following rough outline of the algorithm to be developed herein.

**Outline of the algorithm.** Consider a (linear) CODE as an ordinary differential equation in a special Hilbert space:

$$u'(t) = Au(t) \ , \ u(0) \text{ given.}$$

Discretizing this equation at time $t$ with e.g. the implicit Euler scheme with step size $\tau$ leads to a countable system of algebraic equations (CAE)

$$(I - \tau A) u_{\tau}(t + \tau) = u(t),$$

with $u_{\tau}(t + \tau)$ the Euler approximation at time $t + \tau$ and $u(t)$ given. In order to apply extrapolation in time, we need an asymptotic expansion of the implicit
Euler (or other discretization schemes) in the chosen Hilbert space. This idea has been introduced by Bornemann in [7] and [8] for parabolic partial differential equations. He proves asymptotic expansions for a class of schemes of interest and his results will be applied later in this chapter. If we were able to solve (1.25) exactly, the extrapolation could use order and step-size control as it is well known for ODE's [16]. Fortunately it is shown in [7], that the equation (1.25) has to be solved only up to a tolerance $\epsilon_{CAE}$ for guaranteeing the nice performance of an extrapolation code. Whereas in [7] the equation (1.25) is an elliptic PDE (solved by use of finite elements), in the present context the solution is approximated by use of orthogonal polynomials of a discrete variable. A two parameter family of weight functions $\Psi_{p,a}$ will be introduced allowing the approximation of a wide class of standard distributions (see Chapter 2). The weight functions $\Psi_{p,a}$ lead in a natural way to a scale of Hilbert spaces $H_{p,a}$ and an orthogonal basis of these spaces. In order to solve the CAE (1.25), we employ (as far as possible) analytical properties of the orthogonal polynomials associated to the basis of $H_{p,a}$ called modified discrete Laguerre polynomials. The numerical evaluation of the scalar products (1.24) can be done by a summation algorithm called SUMMATOR. The error control of the SUMMATOR will be obtained adaptively from the requirements of the CAE - solver, in particular the error introduced by the projection into an n-dimensional subspace of $H_{p,a}$ has to dominate the error introduced by the perturbation of this projection. The extrapolation, in turn, prescribes the tolerance $\epsilon_{CAE}$ of the Galerkin method. Thus we get a chain of adaptive requirements:

\[ \epsilon_{INT} \text{ (time-step control)} \rightarrow \epsilon_{CAE} \text{ (Galerkin method)} \rightarrow \epsilon_{SUM} \text{ (SUMMATOR)} \]

1.2 A Two-Parameter Scale of Hilbert Spaces

Motivated by the considerations in Section 1.1 we look for solutions of a CODE in a scale of weighted sequence spaces related to $H_{p}$ from (1.17). These spaces should be spanned by orthogonal functions, which are explicitly known. Furthermore we require good approximation properties for a wide range of problems to be treated. This will be done by adding a parameter $\alpha$.

**Definition 1.1.** Define the weighted sequence spaces $H_{p,a}$ by

\[ H_{p,a} := \left\{ u \in \mathbb{R}^{N} \mid \| u \|_{p,a}^{2} := \sum_{s=1}^{\infty} u_{s}^{2} \Psi_{p,a}(s)^{-1} < \infty \right\}, \]
where the weight function $\Psi_{p,\alpha}(s) > 0$ is given for $s \in \mathbb{N}$ by

$$\Psi_{p,\alpha}(s) = C_{p,\alpha} \left( \frac{s - 1 + \alpha}{s - 1} \right) \rho^{s-1}, \quad 0 < \rho < 1, \quad \alpha > -1,$$

with the constant $C_{p,\alpha} = (1 - \rho)^{1 + \alpha}$ chosen such that $\|\Psi_{p,\alpha}\|_{p,\alpha} = 1$.

**Remark.** Due to the normalization of the $\Psi_{p,\alpha}$, these weight functions will be regarded as probability distributions. It will be pointed out in the introduction of Chapter 2 and in Section 3.2, that the family of distributions $\Psi_{p,\alpha}$ can approximate distributions with sharp peaks as well as broad distributions. In particular for $\alpha = 0$ the weight function $\Psi_{p,0}$ reduces to the geometric distribution

$$\Psi_{p}(s) = (1 - \rho) \rho^{s-1}, \quad 0 < \rho < 1,$$

used in [18], [9] in the context of the discrete Galerkin method. In this case we will omit the parameter $\alpha$. Note that

$$\Psi_{p,\alpha} \in H_{p,\alpha} \quad \text{for} \quad \rho > \rho^2.$$

**Theorem 1.2.**

(i) For $0 < \rho < 1$ and $\alpha > -1$ the space $H_{p,\alpha}$ equipped with the scalar product

$$\langle u, v \rangle_{p,\alpha} := \sum_{s=1}^{\infty} u(s)v(s)\Psi_{p,\alpha}(s)^{-1}, \quad u, v \in H_{p,\alpha},$$

is a separable Hilbert space.

(ii) The embeddings

$$H_{p,\alpha} \hookrightarrow H_{p,\beta} \quad \text{for} \quad 0 < \rho < \rho^* < 1;$$

and

$$H_{p,\alpha} \hookrightarrow H_{p,\beta} \quad \text{for} \quad -1 < \alpha < \beta,$$

are dense and continuous with

$$\|u\|_{p,\beta} \leq (1 - \rho)^{(\alpha - \beta)/2} \|u\|_{p,\alpha}$$

for $u \in H_{p,\alpha}$, $\beta \geq \alpha$ and

$$\|u\|_{p,\alpha} \leq \left( \frac{1 - \rho}{1 - \rho^*} \right)^{(1 + \alpha)/2} \|u\|_{p,\beta}$$

for $u \in H_{p,\alpha}$, $\rho^* < \rho$. 

14
Proof.

(i) This is clear from the theory of weighted sequence spaces (e.g. [11]).

(ii) The inclusions follow from the definitions of the weight function $\Psi_{p,\alpha}$ and the norms $\| \cdot \|_{p,\alpha}$. The estimates (1.32) and (1.33) can be calculated essentially using the definition of the normalizing constants $C_{p,\alpha}$. The embeddings are dense, since the set

$$M := \left\{ u \in \mathbb{R}^\mathbb{N} \mid u_s = 0 \text{ for all but finitely many } s \right\}$$

is dense in all spaces $H_{p,\alpha}$.

In order to clarify the role of the parameters $p$ and $\alpha$, properties of the $H_{p,\alpha}$-spaces will be summarized now.

**Lemma 1.3.** For $0 < \varepsilon < p < 1$ let $u \in H_{p-\varepsilon,0}$. Then for all polynomials $p$ of degree $j$ we have $p \cdot u \in H_{p,\alpha}$ for $\alpha > -1$.

**Proof.** From $u \in H_{p-\varepsilon,0}$ we have by definition

$$\sum_{s=1}^{\infty} u(s)^2 e^{(\lambda+\varepsilon)s} < \infty,$$

if we write $p = e^{-\lambda}$, $\rho - \varepsilon = e^{-\lambda-\varepsilon}$ in terms of $\lambda > 0$, $\varepsilon > 0$. Then there is an $\bar{s} \geq 1$ with

$$e^{(\lambda+\varepsilon)s} > p(s)^2 e^{\lambda s}$$

for all $s > \bar{s}$. Thus $\| pu \|_{p,\alpha}$ is bounded, if

$$\sum_{s=1}^{\infty} u(s)^2 p(s)^2 e^{\lambda s} \leq \sum_{s=1}^{\bar{s}} u(s)^2 p(s)^2 e^{\lambda s} + \sum_{s=\bar{s}+1}^{\infty} u(s)^2 e^{(\lambda+\varepsilon)s}$$

is bounded. This is the case for $u \in H_{p-\varepsilon,0}$. As

$$\left( s - 1 + \alpha \right)^{-1} \leq 1 \text{ for } \alpha \geq 0$$

and

$$\left( s - 1 + \alpha \right)^{-1} \leq \frac{s}{1 + \alpha} \text{ for } -1 < \alpha < 0$$

(1.34)

we can derive $u \in H_{p,\alpha}$ and $pu \in H_{p,\alpha}$ for all $\alpha > -1$.

With Lemma 1.3 we can prove the important
**Corollary 1.4.** If \( u \in H_{p-\epsilon,\alpha} \) for one \( \alpha > -1 \), then \( u \in H_{p,\beta} \) for all \( \beta > -1 \).

The condition \( u \in H_{p-\epsilon,\alpha} \) plays an important role in this work. Under numerical aspects it ensures that we do not approximate an element at the 'edge' of the space \( H_{p,\alpha} \). Corollary 1.4 implies that on this condition the \( p \)-scale is the crucial scale, whereas the \( \alpha \)-scale can be used for approximation purposes.

**Proof of Corollary 1.4.** For \( \beta \geq \alpha \) see Theorem 1.2.(ii). For \( \beta < \alpha \) the estimation
\[
\left( \frac{s - 1 + \alpha}{s - 1} \right) \left( \frac{s - 1 + \beta}{s - 1} \right)^{-1} \leq p(s)^2
\]
holds by use of a polynomial \( p \) with degree \( j \geq (\alpha - \beta)/2 \). Application of Lemma 1.3 leads to the assertion. ■

**Example 1.5.** Let \( \rho \) be given. Define \( u(s) := (\sqrt{\rho})^{s-1} / s \), \( s \geq 1 \). Then
\[
\|u\|_{p,0}^2 = \frac{1}{1 - \rho} \sum_{s=1}^{\infty} \frac{1}{s^2} < \infty ,
\]
but \( \sum_{s=1}^{\infty} u(s) \Psi_{p,0}(s)^{-1} \) is not bounded for any \( \tilde{\rho} < \rho \). For \( v(s) := su(s) \) we see that \( v \notin H_{p,0} \). This confirms, that the assumption of Lemma 1.3 and Corollary 1.4 concerning \( \rho \) is necessary.

**Corollary 1.5.** For \( k = 0, 1, \ldots \), the statistical moments \( \mu_k[u] \) of \( u \) are defined by
\[
(1.35) \quad \mu_k[u] := \sum_{s=1}^{\infty} s^k u(s) .
\]
For \( u \in H_{p,\alpha} \) all statistical moments of \( u \) are bounded.

**Proof.**
\[
\sum_{s=1}^{\infty} s^k u(s) = \sum_{s=1}^{\infty} s^k \Psi_{p,\alpha}(s) u(s) \Psi_{p,\alpha}(s)^{-1} = \left( s^k \Psi_{p,\alpha}, u \right)_{p,\alpha} \leq \|s^k \Psi_{p,\alpha}\|_{p,\alpha} \|u\|_{p,\alpha} = \mu_{2k}[\Psi_{p,\alpha}] \|u\|_{p,\alpha} < \infty ,
\]
using the Cauchy–Schwarz inequality and the fact that all statistical moments of \( \Psi_{p,\alpha} \) are bounded. Corollary 1.5 supplies a nice property of the spaces \( H_{p,\alpha} \) and goes with the considerations in Example 1.2.
The following Lemma 1.6 and Corollary 1.7 concerning the weight function \( \Psi_{\rho,\alpha} \) will be important for the treatment of the convolution operator (1.14) in Example 1.7 later on. We only consider the case \( \alpha = 0 \) here. For \( \alpha \neq 0 \) the constants become a little bit more complicated.

**Lemma 1.6.** For \( 0 < \varepsilon < \rho \) it is \( \Psi_{\varepsilon,0} \in H_{\rho-\varepsilon,0} \) and

\[
(1.36) \quad \| \Psi_{\varepsilon,0} \|_{p-\varepsilon,0} = \frac{1}{\sqrt{\varepsilon}} M_{p,\varepsilon}, \quad M_{p,\varepsilon} := \frac{\sqrt{\rho - \rho + \varepsilon}}{\sqrt{1 - \rho + \varepsilon}}
\]

**Proof.** Omitting the \( \alpha \) – scale in the notation we have

\[
\sum_{s=1}^{\infty} \Psi_{\varepsilon}^2(s)^2 \Psi_{p-\varepsilon}(s)^{-1} = \left( \frac{C_{p-\varepsilon}}{C_{p-\varepsilon}} \right)^2 \sum_{s=1}^{\infty} \left( \frac{\rho - \varepsilon}{\rho} \right)^{s-1}
\]

\[
= \frac{(\sqrt{\rho - \rho + \varepsilon})^2}{\varepsilon(1 - \rho + \varepsilon)} = \frac{1}{\varepsilon} M_{p,\varepsilon}^2,
\]

using the normalizing constant \( C_{\rho} = C_{\rho,0} \) from (1.27).

With Lemma 1.6 we can derive

**Corollary 1.7.** For \( u \in H_{\rho-\varepsilon,0}, \ 0 < \varepsilon < \rho \), the following estimate holds:

\[
(1.37) \quad \sum_{s=1}^{\infty} u(s) \Psi_{\varepsilon,0}(s)^{-1} \leq \frac{1}{\sqrt{\varepsilon}} M_{p,\varepsilon} \| u \|_{p-\varepsilon,0} ,
\]

with a constant

\[
M_{p,\varepsilon} := \frac{(\rho(1 - \rho + \varepsilon))^{1/2}}{1 - \sqrt{\rho}}.
\]

This implies, that the weighted \( l^2 \) norm with parameter \( \rho - \varepsilon \) can be replaced by a weighted \( l^1 \) norm with \( \sqrt{\rho} \) for \( u \in H_{\rho-\varepsilon} \).

**Proof.** Neglecting the \( \alpha \) – dependency again,

\[
\sum_{s=1}^{\infty} u(s) \Psi_{\rho}(s)^{-1} = \frac{C_{p-\varepsilon}}{C_{\sqrt{\rho}} C_{p-\varepsilon}} \sum_{s=1}^{\infty} u(s) \Psi_{p-\varepsilon}(s) \Psi_{p-\varepsilon}(s)^{-1}
\]

\[
\leq \frac{C_{p-\varepsilon}}{C_{\sqrt{\rho}} C_{p-\varepsilon}} \| \Psi_{\varepsilon,0} \|_{p-\varepsilon} \| u \|_{p-\varepsilon} .
\]

Lemma 1.6 and careful insertion of the respective constants \( C_{\rho,0} \) yield the result (1.37).
1.3 THEORY OF COUNTABLE SYSTEMS

Results about existence and uniqueness of solutions of countable systems of ordinary differential equations have been derived by several authors, an extensive survey can be found in the monograph [12] by DEIMLING. He uses a classification of the problem (1.1) into \( s = 1, 2, \ldots \):

(i) lower diagonal systems: 

\[
f_s(t, u_1(t), \ldots) = f_s(t, u_1(t), \ldots, u_s(t)),
\]

(ii) row-finite systems: 

\[
f_s(t, u_1(t), \ldots) = f_s(t, u_1(t), \ldots, u_{\kappa(s)}(t)), \quad \kappa(s) < \infty,
\]

(iii) general systems,

and proves results by means of extensions of ODE techniques (Lipschitz conditions, fixed point iterations). In Section 3.1 we will introduce another classification in the context of the spaces \( H_{p,a} \). Most results in [12] are valid for the classical sequence spaces \( l^p \), \( 1 \leq p \leq \infty \). Linear problems are often considered in terms of infinite matrices and conditions on the matrix entries are derived. This 'matrix point of view' (which is also an 'ODE view') has also been used by other authors, e.g. SHAW [50] for \( l^1 \) and BELLMANN [6] for \( l^p \) spaces. MCLURE/WONG [40] could weaken the conditions for solutions in \( l^1 \) by means of semigroup theory. In [43] solutions of general degradation processes (see Examples 1.2 and 5.3) are obtained by putting strong conditions on the matrix entries. Many further references can be found in [12]. This work will take a different view. The approach suggested herein is motivated by the (expected) qualitative behavior of the solutions. The attention is directed to the (efficient) approximation of solutions of given problems and the spaces \( H_{p,a} \) seem to be promising for this task. In view of the considerations in Section 1.1, we study CODE's only in these spaces as operator equations. It will turn out, that the operators studied in this work are Lipschitz continuous as operators on a fixed \( H_{p,a} \) - space or on the scale of these spaces. For the bounded case we can state the well known result:

**THEOREM 1.8.** For \( 0 \leq t \leq T_f \) let \( A(t) \) be a bounded linear operator on \( H_{p,a} \). Let the function \( t \rightarrow A(t) \) be continuous in the uniform operator topology and \( h(t) \) an continuous \( H_{p,a} \)-valued \( L^1 \)-function. Then for every \( \varphi \in H_{p,a} \) the initial value problem

\[
(1.38) \quad u'(t) = A(t)u(t) + h(t), \quad u(0) = \varphi,
\]

has a unique solution \( u \) in \( H_{p,a} \) for \( t \in [0, T_f] \).
Proof. Theorem 5.1 and p. 129 in [44].

Examples 1.6. (i) Consider the backward difference operator (1.4). An easy computation shows that

\[
\begin{align*}
\|\nabla\|_{p,\alpha} & \leq 1 + \frac{1}{\sqrt{p}}, & \alpha \geq 0, \\
\|\nabla\|_{p,\alpha} & \leq 1 + \frac{1}{\sqrt{p}(1 + \alpha)}, & -1 < \alpha < 0.
\end{align*}
\]

(1.39)

(ii) The norm of the forward difference operator $\Delta$ can be estimated by

\[
\|\Delta\|_{p,\alpha} \leq 1 + M_+
\]

(1.40)

using the norm $M_+$ of the forward shift operator

\[
(S_+ u)(s) := u(s + 1)
\]

given by

\[
\|S_+\|_{p,\alpha} \leq M_+ := \begin{cases} \sqrt{p(1 + \alpha/2)} & \alpha \geq 0, \\
\sqrt{p} & -1 < \alpha < 0. \end{cases}
\]

(1.41)

(iii) A degradation process in polymer chemistry

\[
u'(t) = A_D u(t), \quad u(0) = \varphi,
\]

-- similar to the summatory system (1.7) in Example 1.2 -- can be formulated in terms of the operator $A_D$:

\[
(A_D u)_s := -(s - 1)u_s + 2 \sum_{r=1}^{\infty} u_r \\
= -(s - 1)u_s + 2 \left( \sum_{r=1}^{\infty} (S_+)^r u \right)_s
\]

(1.42)

Using (1.41) the infinite sum of operators on the right-hand side of (1.42) converges uniformly for

\[
\rho(1 + \alpha/2) < 1
\]

(1.43)

(a condition which will play an important role in Example 5.3):

\[
\sum_{r=1}^{\infty} \|S_+\|^r \|\|_{p,\alpha} \leq \sum_{r=1}^{\infty} \|S_+\|^r \|\|_{p,\alpha} \leq \sum_{r=1}^{\infty} M_+^r < \infty.
\]
However, the first term of $A_D$ leads to difficulties. In order to analyze the situation (let $\alpha = 0$) we define an operator $A_1$ by

$$(1.44) \quad (A_1 u)_s := -(s - 1) u_s, \quad s \in \mathbb{N}.$$  

$A_1$ is not bounded in $H_\rho$, but we can derive that

$$(1.45) \quad \|A_1 u\|_\rho \leq \frac{C}{\epsilon} \|u\|_{\rho - \epsilon}, \quad \epsilon > 0,$$

i.e. $A_1$ is Lipschitz continuous as a map from the 'smaller' space $H_{\rho - \epsilon}$ to the 'larger' space $H_\rho$. The solution of

$$(1.46) \quad u'(t) = A_1 u(t), \; u(0) \in H_{\rho - \epsilon},$$

is seen to be

$$u_s(t) = u_s(0) e^{-(s-1)t}, \quad s \in \mathbb{N},$$

and $u(t) \in H_\rho$ for all $t > 0$. An explanation of this behavior can be gained by use of semigroup theory (compare the remark to Corollary 1.12), but a more appropriate description in the present context seems to be Theorem 1.9, which is based on Lipschitz conditions of the form (1.45). Assertion and proof of this result follow Theorem 15.7 in the textbook of DEIMLING [13] and have been converted from certain weighted $l^1$-spaces to the $H_{\rho, \alpha}$-spaces considered here. Additionally, a parameter $\gamma$ has been introduced, which allows the treatment of more general problems than in [13] ($\gamma = 1$). For example, the convolution operator $A_c$ (1.14) requires $\gamma = 1/2$ (Example 1.7), whereas $\gamma = 1$ in (1.45). Theorem 1.9 supplies existence and uniqueness of solutions of nonlinear ODE's in $H_\rho$ (for simplicity the $\alpha$-scale is omitted again and we write $\| \cdot \|_\rho$ instead of $\| \cdot \|_{\rho, \alpha}$).

**Theorem 1.9.** Consider a sub-scale of $H_\rho$-spaces for $\rho \in [\rho_0, 1), \; 0 < \rho_0 < 1$. Let $J = [0, T_f] \subset \mathbb{R}$ and assume:

(a) The operator

$$(1.47) \quad F : J \times H_\rho \longrightarrow H_\rho$$

is continuous for $\bar{\rho} > \rho$ and $F(t, 0) \in H_{\rho_0}$ on $J$.

(b) There exists a constant $M$ such that

$$(1.48) \quad \|F(t, u) - F(t, v)\|_\rho \leq \frac{M}{(\bar{\rho} - \rho)^\gamma} \|u - v\|_\rho, \quad 0 < \gamma \leq 1,$$

for $t \in J$, $\bar{\rho} > \rho$ and $u, v \in H_\rho$.  

---

20
Then for every \( \rho \in (\rho_0, 1) \) the initial value problem

\[ u'(t) = F(t, u(t)), \quad u(0) = \varphi \in H_{\rho_0}, \]

has a unique solution

\[ u : [0, \delta(\bar{\rho} - \rho)^\gamma) \rightarrow H_{\rho_0}, \]

with \( \delta = \min\{T_f, (Md_r)^{-1}\} \). The constant \( d_r > 1 \) can be computed in concrete cases, e.g. \( d_1 = e, \ d_{1/2} = 2\sqrt{3}/3 \).

**Proof.** The proof is based on a fixed point iteration in a scale of Hilbert spaces. We consider the successive approximation

\[ u_k(t) = u_0 + \int_0^t F(s, u_{k-1}(s)) \, ds, \quad k \geq 1, \quad u_0 = \varphi. \]

Due to the condition (a) and \( \varphi \in H_{\rho_0} \), the iterate \( u_k : J \rightarrow H_{\rho_0} \) is continuous for \( \rho > \rho_0 \). We will show by induction, that

\[ \|u_k(t) - u_{k-1}(t)\|_{\rho} \leq C_{\rho_0}(t) \left( \frac{Mtd_r}{(\bar{\rho} - \rho_0)^\gamma} \right)^k \quad \text{for} \quad \bar{\rho} > \rho_0, \]

where \( C_{\rho_0}(t) := \|\varphi\|_{\rho_0} + \frac{(1 - \rho_0)}{M} \max_{s \in [0, t]} \|F(s, 0)\|_{\rho_0}. \)

In a first step we get

\[ \|u_1(t) - u_0\|_{\rho} \leq \int_0^t \|F(s, u_0)\|_{\rho} \, ds \leq \frac{tM}{(\bar{\rho} - \rho_0)^\gamma} C_{\rho_0}(t), \]

using \( \|u\|_{\rho} \leq \|u\|_{\rho_0} \) for \( \rho_0 \leq \bar{\rho} \). Now the induction step yields

\[ \|u_{k+1}(t) - u_k(t)\|_{\rho} \leq \int_0^t \|F(s, u_k(s)) - F(s, u_{k-1}(s))\|_{\rho} \, ds \]

\[ \leq \frac{M}{\varepsilon^\gamma} \int_0^t \|u_k(s) - u_{k-1}(s)\|_{\rho - \varepsilon} \, ds \]

\[ \leq \frac{M}{\varepsilon^\gamma} \int_0^t C_{\rho_0}(t) \frac{(sMd_r)^k}{(\bar{\rho} - \varepsilon - \rho_0)^{\gamma k}} \, ds \]

\[ = \frac{C_{\rho_0}(t)}{\varepsilon^\gamma} \frac{M^{k+1}t^{k+1}d_r^k}{(\bar{\rho} - \varepsilon - \rho_0)^{\gamma k} k + 1}, \]

21
where \( \varepsilon > 0 \) is chosen such that \( \bar{\rho} - \epsilon > \rho_0 \). In order to get rid of the factor \( 1/(k + 1) \) we set

\[
\varepsilon = \frac{\bar{\rho} - \rho_0}{(k + 1)^{1/\gamma}}
\]

and end up with

\[
\|u_{k+1}(t) - u_k(t)\|_\beta \leq C_{\rho_0}(t) \left( \frac{Mtd_\gamma}{(\bar{\rho} - \rho_0)\gamma} \right)^{k+1} \frac{1}{d_\gamma} \left( \frac{k + 1}{((k + 1)^{1/\gamma} - 1)\gamma} \right)^k.
\]

The last factor on the right-hand side is bounded in \( k \) for \( \gamma < 1 \). This leads to the definition of \( d_\gamma \). In particular for \( \gamma = 1/2 \) the maximum is achieved at \( k = 1 \) with \( d_{1/2} \). For \( \gamma = 1 \) the term reduces to \( ((k + 1)/k)^k \leq e \). Thus (1.51) holds for all \( k \) and as \( Mtd_\gamma(\bar{\rho} - \rho)^{-\gamma} < 1 \) for \( t \in [0, \delta(\bar{\rho} - \rho_0)\gamma] \) the sequence \( u_k(t) \) is a Cauchy sequence and converges uniformly on every closed subinterval of \([0, \delta(\bar{\rho} - \rho_0)\gamma]\) to a continuous \( u(t) \) satisfying

\[
u(t) = \varphi + \int_0^t F(s, u(s)) \, ds.
\]

Moreover \( u(t) \) is a solution of the initial value problem, since

\[
F(\cdot, u(\cdot)) : [0, \delta(\bar{\rho} - \rho_0)\gamma] \to H_\beta
\]

is continuous. In order to prove local uniqueness, we consider two solutions \( u(t), v(t) \in H_\beta, \bar{\rho} > \rho_0 \). For fixed \( t \) and \( \rho_1 > \bar{\rho} \),

\[
\|u(t) - v(t)\|_{\rho_1} \leq \frac{Mtd_\gamma}{(\rho_1 - \bar{\rho})\gamma} C_1(t), \quad C_1(t) = \max_{s \in [0,t]} \|u(s) - v(s)\|_{\beta}.
\]

Now, similar to the considerations above, we ‘fill in’ estimates in \( k \) spaces between \( H_\rho \) and \( H_\beta \), setting \( \epsilon = (\rho_1 - \bar{\rho})/(k + 1)^{1/\gamma} \) in the \( k \)-th step — and end up with

\[
\|u(t) - v(t)\|_{\rho_1} \leq C_1(t) \left( \frac{Mtd_\gamma}{(\rho_1 - \bar{\rho})\gamma} \right)^k.
\]

Thus for \( t < (\rho_1 - \bar{\rho})^{-\gamma}(Md_\gamma)^{-1} \) we have \( u(t) = v(t) \). The rest follows by continuation.

**Remarks.** (i) Theorem 1.9 serves only as an example for the structure of the problems we are dealing with. An extension to larger time intervals could be tried as well as the obtainment of results explaining solutions for unbounded operators in a ‘smaller’ space (see \( A_1 \) in Example 1.6). But this might be a
subject of further work.

(ii) The proof of Theorem 1.9 can be changed to get the result for bounded (linear) operators respectively Lipschitz continuous nonlinear operators. Then we can use a standard fixed point iteration in $H_{\rho,\alpha}$.

(iii) The technique which has been used to prove uniqueness could not be extended by the author to get a stability result of the form

$$\| u(t) - v(t) \|_\rho \leq C \phi_\epsilon(Mt) \| u(0) - v(0) \|_{\rho-\epsilon}, \phi_\epsilon \text{ bounded},$$

because the constants in the estimates become unbounded. Such a result might be used for a proof of the convergence of explicit, consistent discretization schemes for operators considered in Theorem 1.9, but we will explain in Section 1.4, why this is not expected.

Example 1.7. To show that Theorem 1.9 is indeed a reasonable tool in our context, we consider the convolution operator $A_C$ (1.14) again. We have to estimate

$$\| A_C(u) - A_C(v) \|_\rho \text{ for } u, v \in H_{\rho-\epsilon}. $$

Denoting by $DA_C(u)$ the Frechet - derivative of $A_C(u)$ with respect to $u$, we can write

$$(DA_C(u) v)(s) = \sum_{r=1}^{s-1} u(s-r) v(r), \ v \in H_{\rho-\epsilon}, \ u \in H_{\rho},$$

and start with

$$\| DA_C(u) v \|_\rho^2 = \sum_{s=1}^\infty \Psi_{\rho}(s)^{-1} \sum_{r=1}^{s-1} u(s-r) v(r) \sum_{k=1}^{s-1} u(s-k) v(k)$$

$$= \sum_{s=1}^\infty v(r) \sum_{k=1}^\infty v(k) \sum_{s=\max(r,k)+1}^{\infty} \Psi_{\rho}(s)^{-1} u(s-r) u(s-k)$$

$$\leq \sum_{s=1}^\infty v(r) \sum_{s=1}^\infty v(k) \sum_{s=1}^{\infty} \Psi_{\rho}(s+r)^{-1} u(s) u(s+r-k)$$

$$+ \sum_{s=1}^\infty v(k) \sum_{s=1}^{\infty} v(r) \sum_{s=1}^{\infty} \Psi_{\rho}(s+k)^{-1} u(s) u(s+k-r)$$

after reordering of the summations. As both terms are of the same structure, we can change the indices $k$ and $r$ in the second term. The inner sum can be expressed by means of the Shift operator $S_+$, after adding

$$|v(k)|\Psi_{\rho}(s+r)^{-1} u_2^* > 0$$

23
to the inner sum of the second term we can rewrite

\[
\| D_A C(u) v \|_p^2 \leq 2 \sum_{r=1}^{\infty} v(r) p^{-r} \sum_{k=1}^{r} |v(k)| (u, S_k^{r-k} u)_p
\]

\[
\leq 2 \| u \|_2^2 \left[ \sum_{r=1}^{\infty} v(r) \sqrt{p^{-r}} \sum_{k=1}^{r} |v(k)| \sqrt{p^{-k}} \right]
\]

\[
\leq 2 \| u \|_2^2 \left( \sum_{r=1}^{\infty} |v(r)| \sqrt{p^{-r}} \right)^2 \leq \frac{1}{c} (1 - p + \epsilon) \| u \|_p^2 \| v \|_p^{2-\epsilon},
\]

with the norm of the shift operator from (1.41) and Corollary 1.7 (using \( \| v \|_{p,\alpha} = ||v||_{p,\alpha} \)). Application of the mean-value theorem gives (1.48) with \( \gamma = 1/2 \).

### 1.4 Discretization and Extrapolation

We apply the idea of a discretization in time of an abstract Cauchy problem in an appropriate Hilbert space. Consider a linear, homogeneous abstract Cauchy problem

\[
(1.52) \quad \frac{du(t)}{dt} = Au(t), \quad u(0) = \varphi.
\]

This differential equation can be discretized for given \( t \) by the \textit{implicit Euler scheme} leading to

\[
(1.53) \quad \frac{u_n \left( \frac{jt}{n} \right) - u_n \left( \frac{(j-1)t}{n} \right)}{\frac{t}{n}} = A u_n \left( \frac{jt}{n} \right), \quad u_n(0) = \varphi,
\]

for \( j = 1, \ldots, n \), which can be written as

\[
(1.54) \quad u_n(t) = \left( I - \frac{t}{n} A \right)^{-n} \varphi.
\]

For bounded operators we can also apply the \textit{explicit Euler scheme} and obtain

\[
(1.55) \quad u_n(t) = \left( I + \frac{t}{n} A \right)^n \varphi.
\]

By \( u_\tau(t) \) we denote the implicit (explicit) Euler approximation (1.54), (1.55) with time step \( \tau = t/n \). For \( A \) a bounded (Lipschitz continuous) operator in a Hilbert space \( H (H = H_{p,\alpha} \) here, of course) we then know
**Theorem 1.10.** (Hairer/Lubich) Given a (possibly) nonlinear CODE

\[(1.56) \quad u'(t) = A(t)u(t), \quad u(0) = \varphi,\]

with \(A\) Lipschitz continuous on \(H\) and unique solution in \(H\). The implicit (explicit) Euler approximation converges for all \(\varphi \in H\) to this solution and the global error of the discretization has an asymptotic expansion

\[(1.57) \quad u_n(t) - u(t) = e_1(t)\tau + \ldots + e_N(t)\tau^N + E_{N+1}(t; \tau)\tau^{N+1}.\]

The \(e_j(t)\) are solutions of certain linear, inhomogeneous differential equations, the remainder \(E_{N+1}(t; \tau)\) is bounded on compact sets.

**Proof.** Because of the boundedness (Lipschitz continuity) of \(A\), we repeat the arguments from ODE theory. By Taylor expansion we conclude that the Euler schemes are consistent with order one. The consistency of the Euler schemes implies the convergence of the schemes analogue e.g to the proof of Theorem 7.3 in [27]. Then the asymptotic expansions follow from [26], Theorem 1.

Things are more difficult for unbounded operators. Assume the explicit Euler scheme applied to a linear equation with operator \(A\), which is bounded in the \(p\)-scale. For ODE's the order of consistency \(p\) is given by

\[(1.58) \quad u(t + \tau) - u(t) - \tau A u(t) = O(\tau^{p+1}).\]

The question is, in what sense we can speak of consistency now. Let \(\rho\) be given and \(u(0) \in H_{\rho_0}, \rho_0 < \rho\). Assuming the validity of a Lipschitz condition (1.48), we know from Theorem 1.9 that (1.58) is defined in \(H_\rho\) for \(t\) small enough, leading to \(p = 1\) for the explicit Euler scheme. Even the expression \(u_n(t)\) from (1.55) is defined in \(H_\rho\) by stepping in \(n\) spaces between \(H_{\rho_0}\) and \(H_\rho\). But an estimation which allows the limit \(n \to \infty\) is not possible here, because the insertion of infinite many spaces between \(H_{\rho_0}\) and \(H_\rho\) leads to unbounded Lipschitz constants. As mentioned in Remark (iii) to Theorem 1.9, the same reason prevents the proof of a stability result, which possibly could be applied to prove convergence of consistent discretization schemes as in standard ODE theory. As a consequence we can use the explicit Euler scheme only for the bounded case.

The examination of the implicit Euler discretization requires assumptions which can be naturally formulated in terms of semigroup theory. We only study linear (homogeneous) problems in detail. Nonlinear equations will be attacked by a *semi-implicit discretization* (see Chapter 4.1, (6.1)), which is the implicit Euler scheme in the linear case, of course.
Strongly continuous semigroups. A one parameter family \( T(t) \), \( 0 \leq t \leq \infty \) of bounded linear operators from a Banach space \( X \) into \( X \) is called a semigroup, if

\[
\begin{align*}
(i) \quad & T(0) = I, \ (I \ \text{Identity on} \ X), \\
(ii) \quad & T(t + t') = T(t)T(t') \quad \text{for every} \ t, t' \geq 0.
\end{align*}
\]

The linear operator \( A \) defined by

\[
\begin{align*}
(i) \quad & D(A) := \left\{ x \in X \mid \lim_{t \to 0} \frac{T(t)x - x}{t} \ \text{exists} \right\} \\
(ii) \quad & Ax := \lim_{t \to 0} \frac{T(t)x - x}{t} = \frac{d}{dt}T(t)x \bigg|_{t=0}, \ x \in D(A),
\end{align*}
\]

is called the infinitesimal generator of the semigroup \( T(t) \) with domain \( D(A) \). In the following we will consider strongly continuous or \( C_0 \) - semigroups which additionally have the property

\[
\begin{align*}
(iii) \quad & \lim_{t \to 0} T(t)x = x \quad \text{for every} \ x \in X.
\end{align*}
\]

For such semigroups the abstract Cauchy problem (1.52) has a unique solution for \( \varphi \in D(A) \) and the solution is given by

\[
u(t) = T(t)\varphi. \]

It has been shown in [7] that for homogeneous abstract Cauchy problems (1.52) with \( A \) generator of a \( C_0 \) - semigroup the global error of the implicit Euler scheme has an asymptotic expansion. As we want to apply extrapolation, we have to ensure that the semigroup approach is reasonable for CODE’s formulated in \( X = H_{\rho,a} \).

For a \( C_0 \) semigroup there are constants \( \omega \geq 0 \) and \( M \leq 1 \) such that \( \|T(t)\| \leq Me^{\omega t} \) for \( t \geq 0 \). If \( \omega = 0 \) and \( M = 1 \), \( T(t) \) is called a \( C_0 \) semigroup of contractions. A characterization of the infinitesimal generator of such a semigroup is given by the HILLE - YOSIDA theorem.

**Theorem 1.11.** A linear (possibly) unbounded operator \( A \) in a Banach space \( X \) is the infinitesimal generator of a \( C_0 \) semigroup of contractions \( T(t), t \geq 0 \), if and only if

(i) \( A \) is closed and \( D(A) \) is dense in \( X \).

(ii) The resolvent set \( \rho(A) \) of \( A \) contains \( \mathbb{R}^+ \) and for every \( \lambda > 0 \)

\[
\|R(\lambda; A)\| \leq \frac{1}{\lambda},
\]

where \( R(\lambda; A) := (\lambda I - A)^{-1} \) is the resolvent of \( A \).
Proof. See e.g. the textbook of PAZY [44], Theorem 3.1.

Remark. Characterizations of other $C_0$-semigroups can be found in [44], Corollary 3.8 ($||T(t)|| \leq e^{\omega t}$) and Theorem 5.2 ($||T(t)|| \leq M$). For a general $C_0$-semigroup (1.62) is replaced by ([44], Theorem 5.3):

$$||R(\lambda; A)^n|| \leq \frac{M}{(\lambda - \omega)^n}, \quad \text{for } \lambda > \omega, \ n = 1, 2, \ldots.$$

COROLLARY 1.12. The operator $A_1$ from (1.44) is the infinitesimal generator of a $C_0$ semigroup of contractions in $H_{\rho,\alpha}$.

Proof. It is easily seen that

$$H_{\rho,\alpha} \subset D(A) = \left\{ u \in H_{\rho,\alpha} \mid \sum_{s=1}^{\infty} (s - 1)^2 u(s)^2 \Psi_{\rho,\alpha}^{-1} < \infty \right\}$$

for $\epsilon > 0$ and therefore $D(A)$ is dense in $H_{\rho,\alpha}$. For $u \in D(A)$

$$||(\lambda I - A_1)u||_{\rho,\alpha} \geq \lambda \|u\|_{\rho,\alpha}, \ \lambda > 0,$$

holds, so $A_1$ is dissipative. Finally we have for $u \in H_{\rho,\alpha}$

$$(I - A_1) \frac{u(s)}{1 + s} = u(s),$$

where

$$\frac{u(s)}{1 + s} \in H_{\rho,\alpha}.$$

By the LUMER-PHILLIPS Theorem ([44], Theorem 4.3) the assertion follows.

Remark. With $A_1$ generator of a $C_0$-semigroup, we have another proof, that the initial value problem (1.46) has a solution in $H_{\rho}$. Instead of Theorem 1.9 in Section 1.3, we apply Theorem 1.2.4 in [44].

An asymptotic expansion of the implicit Euler scheme is given now by

THEOREM 1.13. (Bornemann) Let $A$ be the infinitesimal generator of a $C_0$ semigroup of contractions on a Banach space $X$ and let

$$\varphi \in D(A^{2N+2}).$$
Then for the global error of the implicit Euler scheme the asymptotic expansion

\[ u_+(t) - u(t) = e_1(t)\tau + \ldots + e_N(t)\tau^N + E_{N+1}(t;\tau)\tau^{N+1} \]

is valid in \( X \). Furthermore the estimates

\[ \begin{align*}
& a) \quad \|e_k(t)\| \leq Ct, \quad 1 \leq k \leq N, \\
& b) \quad \|E_{N+1}(t;\tau)\| \leq Ct
\]  

hold for every \( t \in [0,T_f] \), \( \tau \in [0,\tau_0] \). The constant \( C \) depends only on \( T_f, \tau_0, N \) and \( \varphi \). For general \( C_0 \) semigroups there is an additional time-step bound \( \tau < 1/\omega \).

**Proof.** See [7], Theorem 2.13.

In the present context, the important condition \( \varphi \in D(A^{2N+2}) \) characterizing the consistency of initial data can be gained from

**COROLLARY 1.14.** Let \( A \) be an operator satisfying the conditions of Theorem 1.9 in \( H_{p,\alpha} \) and let \( \varphi \in H_{p-\varepsilon,\alpha} \). Then \( \varphi \in D(A^N) \) for all fixed \( N \in \mathbb{N} \).

**Proof.** Define \( \varepsilon_k := \varepsilon/k \), \( 1 \leq k \leq N \). Then \( A \) is bounded from \( H_{p-\varepsilon_k,\alpha} \) to \( H_{p-\varepsilon_{k+1},\alpha} \). \( \blacksquare \)
2 Modified Discrete Laguerre Polynomials

The experiences made with the discrete Galerkin method in [18], [9], [52] and the considerations in Chapter 1 admit the expectation, that the weight function \( \Psi_{p,\alpha} \) will be a good basis for the Galerkin approximation of realistic problems. For an illustration we discuss the behavior of \( \Psi_{p,\alpha} \) for different values of \( p \) and \( \alpha \).

For \( \alpha = 0 \) the weight function is reduced to the geometric distribution, which plays an important role in polymer chemistry under the name Schulz-Flory distribution [45], [9]. For further discussions we need the first statistical moments of \( \Psi_{p,\alpha} \). The constant \( C^{p,\alpha} \) from (1.27) is chosen such that

\[
(2.1) \quad \nu_0(p, \alpha) = \nu_0 := \mu_0[\Psi_{p,\alpha}] = 1, \quad 0 < p < 1, \quad \alpha > -1.
\]

Using the definition of the binomial coefficient, we can derive that

\[
\nu_1(p, \alpha) := \mu_1[\Psi_{p,\alpha}] = 1 + \sum_{s=2}^{\infty} (s - 1) \Psi_{p,\alpha} =
\]

\[
= 1 + \rho \sum_{s=1}^{\infty} (s + \alpha) \Psi_{p,\alpha} = 1 + \rho \alpha + \rho \nu_1(p, \alpha)
\]

and therefore

\[
(2.2) \quad \nu_1(p, \alpha) = \frac{1 + \alpha \rho}{1 - \rho}.
\]

By a similar procedure we get

\[
(2.3) \quad \nu_2(p, \alpha) := \mu_2[\Psi_{p,\alpha}] = \frac{\alpha^2 \rho^2 + 3 \alpha \rho + \rho + 1}{(1 - \rho)^2}.
\]

The so-called dispersion coefficient \( d \) is defined by

\[
(2.4) \quad d := \frac{\mu_0 \nu_2}{\mu_1^2}
\]

and can give some hints about the weight function for special settings of \( p \) and \( \alpha \). For \( \alpha \gg 1 \) \( d \) tends to 1, which means that \( \Psi_{p,\alpha} \) becomes a very peaked distribution. In fact, if we set \( \alpha = \lambda / \rho \), \( \lambda \) the parameter of a Poisson distribution, \( \Psi_{p,\alpha} \) converges pointwise to the Poisson distribution for \( \rho \to 0 \). A motivation of such a parameter setting will be given at the end of Section 3.2. Therewith the two discrete Galerkin methods suggested in [18], based on the geometric distribution (discrete Laguerre polynomials) and the Poisson distribution (Charlier polynomials) are combined by the two-parameter family \( \Psi_{p,\alpha} \). We note, that a Galerkin method based on the Poisson distribution itself
would fail in our context, because the application of Euler steps destroys the strong decaying behavior for \( s \to \infty \) of elements of the associated sequence space, implying that Euler steps cannot be performed.

Now we turn to broad distributions \((d \gg 1)\) which can be approximated well by choosing \( \alpha < 0 \) and \( \rho \) close to one. Then \( \Psi_{\rho,\alpha} \) approaches a hyperbola of the form \( 1/s^\alpha \). This has been used in [52] to model fractional powers appearing in certain chemical processes. The connection between both extreme properties \((d \to 1 \text{ and } d \gg 1)\) with only one family of weight functions is the key to the efficiency of the method to be developed herein.

In this chapter, an orthogonal basis of the spaces \( H_{\rho,\alpha} \) introduced in Section 1.2 will be constructed. This basis is given by means of the modified discrete Laguerre polynomials which will be derived in Section 2.1. Important properties and transformation formulas of these polynomials are added in Section 2.2. The approximation of an element \( u \in H_{\rho,\alpha} \) by the orthogonal basis is backed by a theorem proven in Section 2.3. As not stated otherwise, we always assume \( 0 < \rho < 1 \text{ and } \alpha > -1 \).

## 2.1 Construction of the Polynomials

In this section we will construct first a set of orthogonal polynomials \( \{l_k\} \) with respect to the scalar product

\[
(u, v)^{\rho,\alpha} := \sum_{s=1}^{\infty} u(s) v(s) \Psi_{\rho,\alpha}(s),
\]

where \( u, v : \mathbb{N} \to \mathbb{R} \) can be interpreted as sequences or as grid functions on \( \mathbb{N} \). The isometric isomorphism

\[
T_{\rho,\alpha} : H^{\rho,\alpha} := \left\{ u \in \mathbb{R}^\mathbb{N} \mid (u, u)^{\rho,\alpha} < \infty \right\} \longrightarrow H_{\rho,\alpha}
\]

defined by

\[
(T_{\rho,\alpha} u)(s) = u(s) \Psi_{\rho,\alpha}(s)
\]

will transform the polynomial basis \( \{l_k(\rho, \alpha)\} \) of \( H^{\rho,\alpha} \) to the basis \( \{\psi_k(\rho, \alpha)\} := \{ \Psi_{\rho,\alpha} l_k(\rho, \alpha) \} \) of \( H_{\rho,\alpha} \). This has to be kept in mind in the following, because it will be switched between both bases very often.

We know from [10], Theorem 3.1, that there exists an orthogonal set of polynomials with respect to the scalar product (2.5), if all statistical moments \( \mu_k[\Psi_{\rho,\alpha}] \) are bounded and the determinant condition

\[
\Delta_n := \det (\mu_{i+j}[\Psi_{\rho,\alpha}])_{i,j=0}^{n} = \begin{vmatrix} \mu_0 & \mu_1 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_n & \mu_{n+1} & \cdots & \mu_{2n} \end{vmatrix} \neq 0
\]

30
holds for \( n = 0, 1, \ldots \). Moreover, in this case the associated orthogonal polynomials can be generated by a three-term-recurrence formula. We will not check these conditions for the special weight function \( \Psi_{\rho, \alpha} \) here, since the construction of the corresponding orthogonal polynomials of a discrete variable is well known in the literature. For example, the discrete Laguerre polynomials associated to the geometric distribution have been studied by Gottlieb [25] in 1938. Charlier has described the polynomials named after him in 1906. For the modified discrete Laguerre polynomials we will follow the considerations of Lesky [36], [37] and the textbook of Nikiforov and Uvarov [42], which gives a modern survey about orthogonal polynomials. For a group theoretical classification we refer to the work of Koornwinder (e.g. [35]). Properties of discrete orthogonal polynomials are given by Askey and Gasper [1], [2], [24] and others. This is only a sketchy account of the abundant literature in the field. Thus some of the results proven in this chapter (in particular in Section 2.2) may already be known to the experts.

A standard technique for the construction of the classical (continuous) orthogonal polynomials is the use of so-called Rodrigues formulas. It has been shown e.g in [36] and [42], that for some types of discrete weight functions there are discrete Rodrigues formulas. A classification of these weight functions can be found in [10], where generating functions are the starting point. In [42] discrete hypergeometric equations are considered, whereas in [36] a variational approach is chosen. In this section, we begin with the proof of a given Rodrigues formula. For ease of writing sometimes the forward product

\[
(a)_n := a (a + 1) \ldots (a + n - 1), \quad a \in \mathbb{R},
\]

will be used.

**Theorem 2.1.** The Rodrigues formula

\[
(2.9) \quad l_n(s; \rho, \alpha) = \frac{(1 + \alpha)_n}{n!} \Psi_{\rho, \alpha}(s)^{-1} \Delta^n \left\{ C_{\rho, \alpha} \rho^{s-1} \left( \frac{s - 1 + \alpha}{s - 1 - n} \right) \right\}
\]

generates polynomials \( l_n(s) = l_n(s; \rho, \alpha) \) which are orthogonal with respect to the scalar product \((2.1)\) for \( 0 < \rho < 1 \) and \( \alpha > -1 \). The orthogonality relation reads

\[
(2.10) \quad \langle l_m, l_n \rangle_{\rho, \alpha} = \delta_{nm} \gamma_{n, \alpha}^{\rho, \alpha}, \quad \gamma_{n, \alpha}^{\rho, \alpha} := \rho^n \left( \frac{n + \alpha}{n} \right).
\]

The polynomials \( l_n(s; \rho, \alpha) \) will be called modified discrete Laguerre polynomials.
Proof. First we evaluate the difference on the right-hand side of (2.9).
\[
\Delta^n \left\{ C_{p,a} \rho^{s-1} \left( \frac{s-1 + \alpha}{s-1 - n} \right) \right\} = C_{p,a} \sum_{k=0}^{n} \binom{n}{k} (-1)^{n-k} \rho^{s-1+k} \left( \frac{s-1 + k + \alpha}{s-1 + k - n} \right)
\]
using the binomial theorem after representing \( \Delta^n = (S_+ - I)^n \) in terms of the forward shift operator \( S_+ \) and the identity operator \( I \). As
\[
\left( \frac{s-1 + k + \alpha}{s-1 + k - n} \right) = \left( \frac{s-1 + \alpha}{s-1} \right) \frac{(s-1 + k + \alpha) \ldots (s + \alpha)(s-1) \ldots (s - n + k)}{(1 + \alpha) \ldots (n + \alpha)}
\]
we obtain
\[
(2.11) \quad l_n(s) = \sum_{k=0}^{n} (-1)^{n-k} \rho^k \left( s-1 + \alpha \right) \binom{s-1}{n-k} \left( \frac{s-1}{n-k} \right)
\]
which obviously is a polynomial of degree \( n \).

In order to show that
\[
(l_m, l_n)^{p,a} = 0 \quad \text{for} \quad m \neq n,
\]
assume \( n > m \) (for \( m > n \) the argumentation remains the same).

(2.12) \quad (l_m, l_n)^{p,a} = C_{p,a} \frac{(1 + \alpha)_n}{n!} \sum_{s=1}^{\infty} l_m(s) \Delta^n \left\{ \rho^{s-1} \left( \frac{s-1 + \alpha}{s-1 - n} \right) \right\},
\]
where the Rodrigues formula has been inserted. From [42], p. 107, an iterated summation by parts formula
\[
\sum_{s=A}^{B} f(s) \Delta^n g(s) = (-1)^n \sum_{s=A}^{B} g(s+n) \Delta^n f(s)
\]
\[
(2.13) \quad + \sum_{k=0}^{n-1} (-1)^k \Delta^k f(s) \Delta^{n-1-k} g(s+k) \Bigg|_{s=A}^{s=B}
\]
can be derived for finite or infinite summation bounds \( A \) and \( B \). Application of this formula to (2.12) yields
\[
(l_m, l_n)^{p,a} = \left( -1 \right)^n C_{p,a} \frac{(1 + \alpha)_n}{n!} \sum_{s=1}^{\infty} \rho^{s-1+n} \left( \frac{s-1+\alpha+n}{s-1} \right) \Delta^n l_m(s) +
\]
\[
+ C_{p,a} \frac{(1 + \alpha)_n}{n!} \sum_{k=0}^{n-1} (-1)^k \Delta^k l_m \Delta^{n-1-k} \left\{ \rho^{s-1+k} \left( \frac{s-1+k+\alpha}{s-1+k-n} \right) \right\} \Bigg|_{s=1}^{\infty}.
\]
(2.14)
The first sum is equal to zero since $\Delta^n l_m(s) = 0$ for $n > m$, in the second sum we look at the difference expression

$$
\Delta^{n-1-k} \left\{ \rho^{s-1-k} \left( \frac{s-1+k+\alpha}{s-1+k-n} \right) \right\} =
\sum_{\nu=0}^{n-1-k} \left( \frac{n-1-k}{\nu} \right) (-1)^{n-1-k-\nu} \rho^{s-1+\nu+\alpha} \left( \frac{s-1+k+\nu}{s-1+k+\nu-n} \right).
$$

The terms in this sum tend to zero for $s \to \infty$, because $0 < \rho < 1$. For $s = 1$ we have to consider $\left( \frac{k+\nu}{k+\nu-n} \right)$, which is zero for $\nu \leq n-1-k$ and $\alpha > -1$.

Now we compute the orthogonality factors

$$\gamma_n^{\rho,\alpha} = (l_n, l_n)^{\rho,\alpha}.$$

For $n = m$ the second sum in (2.14) again can be seen to be zero, whereas for the evaluation of the first sum we need the difference $\Delta^n l_n(s)$. We use the fact that

$$\Delta^n s^k = 0 \quad \text{for} \quad k < n \quad \text{and} \quad \Delta^n s^n = n!,$$

implying that in the series representation (2.11) only the coefficient of $s^n$ has to be considered, so that

(2.15) \quad $\Delta^n l_n(s) = (\rho - 1)^n$.

From the normalization (2.1) of the weight function $\Psi_{\rho,\alpha}$ we know that

(2.16) \quad $\sum_{s=1}^{\infty} \rho^{s-1} \left( \frac{s-1+\alpha+n}{s-1} \right) = \frac{1}{(1-\rho)^{1+\alpha+n}}$.

Summarizing (2.14), (2.15), (2.16) we get the result

(2.17) \quad $(l_n, l_n)^{\rho,\alpha} = \rho^n \left( 1+\alpha \right)_n = \rho^n \left( \frac{n+\alpha}{n} \right)$.

The modified discrete Laguerre polynomials can be regarded as special Meixner polynomials. This can be seen by writing

$$\left( \frac{s-1+\alpha}{s-1} \right) = \frac{\Gamma(s+\alpha)}{\Gamma(s) \Gamma(1+\alpha)}$$

in terms of the Gamma–function $\Gamma$. In [42] the Meixner polynomials are shown to be orthogonal with respect to the weight function

$$\Psi(s) = \frac{\rho^s \Gamma(\gamma+s)}{\Gamma(1+s) \Gamma(\gamma)} \quad \gamma > 0 \quad 0 < \rho < 1 \quad s = 0, 1, \ldots.$$
It is easily seen that this are in principle the modified discrete Laguerre polynomials with another normalization and a different lower summation bound in the scalar product. In fact, the \( l_n(s; \rho, \alpha) \) are connected with the Meixner polynomials \( m_n(s; \rho, \gamma) \) as given in [24] by

\[
(2.18) \quad l_n(s; \rho, \alpha) := \frac{\rho^n}{n!} m_n(s - 1; \rho, 1 + \alpha),
\]

which in turn can be expressed by means of the hypergeometric function

\[
(2.19) \quad {}_2F_1(a, b; c; z) := \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k k!} z^k,
\]

such that

\[
(2.20) \quad m_n(s; \rho, \gamma) = (1 + \alpha)_n \, {}_2F_1(-n, -s, \gamma; 1 - \frac{1}{\rho}).
\]

For \( \gamma = 1 + \alpha \) and with \( s \to (s - 1) \) thus we have

\[
(2.21) \quad l_n(s; \rho, \alpha) = \rho^n \frac{(1 + \alpha)_n}{n!} \sum_{k=0}^{n} \frac{(-n)_k (-s + 1)_k}{(1 + \alpha)_k k!} \left(1 - \frac{1}{\rho}\right)^k = \sum_{k=0}^{n} \rho^{n-k} (\rho - 1)^k \binom{n+\alpha}{n-k} \frac{(s-1)^{k}}{k!}
\]

This series representation is more convenient than (2.11) and will be used in Section 2.2.

There are many properties of the several polynomials mentioned above, but we restrict ourselves to properties to be applied in the present context. Additional informations about Meixner and related polynomials can be found in the literature referred to above. The modified discrete Laguerre polynomials with \( \alpha = 0 \) have been used for the solution of CODE’s in [18] and [9]. In [52], the parameter \( \alpha \) already has been introduced in order to realize a discrete Galerkin method for certain so-called heterogeneous processes. Contrary to that, in this work the parameter \( \alpha \) is used for extending the approximation properties of the suggested approach. An application in statistics (not associated to the discrete Galerkin method) is presented in [41].

An important tool for the generation of orthogonal polynomials is the three-term-recurrence relation. For the modified discrete Laguerre polynomials the recurrence is started with \( l_{-1} = 0 \) and \( l_0 = 1 \). Using a relation for Meixner polynomials in [42], we obtain with (2.18):

\[
(2.22) \quad (n + 1)l_{n+1}(s; \rho, \alpha) = [(n + \alpha + 1)\rho + n - (1 - \rho)(s - 1)]l_n(s; \rho, \alpha) - (n + \alpha)\rho l_{n-1}(s; \rho, \alpha).
\]
Finally a fundamental difference formula is taken from [42], serving as a basic relation in Section 2.2:

\[(2.23) \quad l_n(s+1;\rho,\alpha) - l_n(s;\rho,\alpha) = (\rho - 1) l_{n-1}(s;\rho,\alpha + 1).\]

Note that by (2.23) the forward difference operator induces a shift in the \(\alpha\) scale.

### 2.2 Basic Properties of the Polynomials

The approximation of solutions of infinite dimensional linear systems in a most suited space \(H_{\alpha,\alpha}\) requires transformations between the polynomial sets \(\{l_j(s;\rho,\alpha)\}\) and \(\{l_j(s;\bar{\rho},\alpha)\}\) for \(\alpha \neq \bar{\alpha}, \rho \neq \bar{\rho}\). Moreover, the weight function fitting condition derived in Section 3.2 has to be fulfilled. As it is more convenient, the transformation will be done for both parameters separately. We begin with the parameter \(\rho\).

**Lemma 2.2.** The transformation between \(l_j(s;\rho,\alpha)\) and \(l_j(s;\bar{\rho},\alpha)\) can be expressed by

\[
(2.24) \quad l_j(s;\rho,\alpha) = \sum_{k=0}^{j} d_{\alpha}^{j,k}(\rho, \bar{\rho}) l_k(s;\rho,\alpha), \quad 0 < \bar{\rho} < 1 ,
\]

\[
d_{\alpha}^{j,k}(\rho, \bar{\rho}) := \frac{(\bar{\rho} - \rho)^{j-k}}{(1-\rho)^{j}} (1 - \bar{\rho})^{k} \frac{\binom{j + \alpha}{k}}{\binom{j-k}{k}}, \quad j \geq k \geq 0 .
\]

**Proof.** After inserting the series representation (2.21) of the \(l_k(s;\rho,\alpha)\) and the definition of the \(d_{\alpha}^{j,k}(\rho, \bar{\rho})\) a reordering of the summations leads to

\[
\sum_{k=0}^{j} d_{\alpha}^{j,k}(\rho, \bar{\rho}) l_k(s;\rho,\alpha) = \left(\frac{\bar{\rho} - \rho}{1 - \rho}\right)^{j} \sum_{k=0}^{j} \left(\frac{\rho - 1}{\rho}\right)^{k} \binom{j+\alpha}{j-k} \times
\]

\[
\times \sum_{\nu=0}^{j-k} \left(\frac{\nu}{\nu-k}\right) (\bar{\rho} - \rho)(1 - \bar{\rho})^{\nu} .
\]

With the relation ([46],p. 3)

\[
(2.25) \quad \binom{j}{\nu} \binom{\nu}{k} = \binom{j}{k} \binom{j-k}{\nu-k}
\]

and the Binomial theorem we end up (after a few manipulations) with

\[
\sum_{k=0}^{j} d_{\alpha}^{j,k}(\rho, \bar{\rho}) l_k(s;\rho,\alpha) = \sum_{k=0}^{j} \left(\frac{j + \alpha}{j-k}\right) \left(\frac{s-1}{k}\right) (\bar{\rho} - 1)^{k} \bar{\rho}^{j-k} = l_j(s;\bar{\rho},\alpha)
\]
As a consequence of this lemma we notice that

\[
\frac{1}{\gamma_{j,k}^\alpha}(l_j(\bar{\rho}, \alpha), l_k(\rho, \alpha))_{\rho, \alpha} = d_{\alpha}^j k(\rho, \bar{\rho})
\]

for \( k \leq j \). For \( k > j \) the scalar product is zero, of course.

**Corollary 2.3.** An element \( u(s) \in H_{\rho, \alpha} \) with basis expansion

\[
u(s) = \sum_{j=0}^{\infty} a_j(\rho, \alpha) \psi_j(s; \rho, \alpha) = \Psi_{\rho, \alpha} \sum_{j=0}^{\infty} a_j(\rho, \alpha) l_j(s; \rho, \alpha),
\]

which is also in the space \( H_{\rho, \alpha} \), can be expressed in the \( H_{\rho, \alpha} \)-basis with the coefficients

\[
a_j(\bar{\rho}, \alpha) = \frac{1}{\rho^j} \sum_{k=0}^{\infty} a_k(\rho, \alpha) \binom{j}{k} (\bar{\rho} - \rho)^{j-k} (1 - \rho)^j (1 - \rho)^k.
\]

Note that this transformation is independent of \( \alpha \).

**Proof.** The projection of \( u \) to a basis element \( \psi_j(\bar{\rho}, \alpha) = \Psi_{\rho, \alpha} l_j(\bar{\rho}, \alpha) \) can be written as

\[
\frac{1}{\gamma_{j}^{\rho, \alpha}} (u, \psi_j(\bar{\rho}, \alpha))_{\rho, \alpha} = \frac{1}{\gamma_{j}^{\rho, \alpha}} \sum_{k=0}^{\infty} a_k(\rho, \alpha) (l_k(\rho, \alpha), l_j(\bar{\rho}, \alpha))_{\rho, \alpha}
\]

Insertion of Lemma 2.2 in the form (2.26) and remembering in the definition of the \( \gamma_{j}^{\rho, \alpha} \) leads to (2.27).

**Corollary 2.4.** The derivative of the \( l_j(s; \rho, \alpha) \) with respect to \( \rho \) can be expressed by

\[
\frac{\partial l_j(s; \rho, \alpha)}{\partial \rho} = \frac{(j + \alpha) l_{j-1}(s; \rho, \alpha) - j l_j(s; \rho, \alpha)}{1 - \rho}.
\]

**Proof.** Evaluation of the expression

\[
\frac{l_j(s; \rho + \Delta \rho, \alpha) - l_j(s; \rho, \alpha)}{\Delta \rho} = \frac{1}{\Delta \rho} \left[ \sum_{i=0}^{j} d_{\alpha}^i (\rho, \rho + \Delta \rho) l_i(s; \rho, \alpha) - l_j(s; \rho, \alpha) \right]
\]

for \( \Delta \rho \to 0 \).

Now the transformation formulas for varying \( \alpha \) will be derived.
Lemma 2.5. The transformation between \( l_j(s; \rho, \alpha) \) and \( l_j(s; \rho, \bar{\alpha}) \) is given by

\[
\begin{align*}
l_j(s; \rho, \alpha) &= \sum_{k=0}^{j} d_{\rho}^{j,k}(\alpha, \bar{\alpha}) l_k(s; \rho, \alpha), \quad \bar{\alpha} > -1, \\
d_{\rho}^{j,k}(\alpha, \bar{\alpha}) &= \rho^{j-k} \binom{\alpha - \alpha + j - k - 1}{j - k}, \quad j \geq k \geq 0.
\end{align*}
\]

Proof. Following the lines of the proof of Lemma 2.2 we insert the series expansion of \( l_j(s; \rho, \alpha) \) into the series on the right-hand side of \( (2.29) \) and reorder the summations:

\[
\sum_{k=0}^{j} d_{\rho}^{j,k}(\alpha, \bar{\alpha}) l_k(s; \rho, \alpha) = \rho^{j} \sum_{\nu=0}^{j} \binom{\rho - 1}{\nu} \binom{s - 1}{\nu} \sum_{k=\nu}^{j} \binom{k + \alpha}{k - \nu} \binom{\bar{\alpha} - \alpha + j - k - 1}{j - k}.
\]

If we can show that the inner sum is equal to \( \binom{j+\alpha}{j-\nu} \) we will be finished. By setting \( m = j - \nu, n = \bar{\alpha} - \alpha + j - \nu, p = \nu + \alpha \), we can use a relation from [46], p. 10, which is given there only for natural numbers but can be extended easily to real \( n \) and \( p \). Then

\[
\sum_{k=0}^{m} \binom{p + k}{k} \binom{n - k - 1}{m - k} = \binom{n + p}{m} = \binom{j + \bar{\alpha}}{j - \nu}.
\]

The transformation of coefficients \( a_j(\rho, \alpha) \) of an \( H_{\rho,\alpha} \)-basis expansion to coefficients \( a_j(\rho, \bar{\alpha}) \) of an \( H_{\rho,\bar{\alpha}} \)-expansion works analogue to Corollary 2.3 and is independent of \( \rho \) in this case. This nice feature shows how the two scales are separated. A differentiation formula can be derived similar to Corollary 2.4:

Corollary 2.6. The derivative of the \( l_j(s; \rho, \alpha) \) with respect to \( \alpha \) can be expressed by

\[
\frac{\partial l_j(s; \rho, \alpha)}{\partial \alpha} = \sum_{\nu=0}^{j} \frac{\rho^{j-\nu}}{(j-\nu)} l_\nu(s; \rho, \alpha).
\]

Proof. Application of the definition of the binomial coefficients. ■

Now we prove two important shift properties of the discrete Laguerre polynomials.

37
COROLLARY 2.7. The application of the forward difference operator to \( l_j(s; \rho, \alpha) \) can be expanded into

\[
\Delta l_j(s; \rho, \alpha) = l_j(s + 1; \rho, \alpha) - l_j(s; \rho, \alpha) = (\rho - 1) \sum_{k=0}^{j-1} \rho^{j-1-k} l_k(s; \rho, \alpha) .
\]

(2.31)

Proof. Insertion of Lemma 2.2 into the fundamental difference relation (2.23):

\[
l_j(s + 1; \rho, \alpha) - l_j(s; \rho, \alpha) = (\rho - 1) l_{j-1}(s; \rho, \alpha + 1)
\]

leads to transformation coefficients \( d^p_k(\alpha, \alpha + 1) = \rho^{j-1-k} \).

COROLLARY 2.8. The application of the backward difference operator to \( l_j(s; \rho, \alpha) \) can be expressed by

\[
(2.32) \quad \nabla l_j(s; \rho, \alpha) = l_j(s - 1; \rho, \alpha) - l_j(s; \rho, \alpha) = (1 - \rho) \sum_{k=0}^{j-1} l_k(s; \rho, \alpha) .
\]

Proof. Backward shift of the difference relation (2.31) in the argument \( s \) yields

\[
l_j(s - 1; \rho, \alpha) + (\rho - 1) \sum_{k=0}^{j-1} \rho^{j-1-k} l_k(s - 1; \rho, \alpha) = l_j(s; \rho, \alpha) ,
\]

which can be regarded as an infinite triangular system of linear equations in the variables \( l_j(s - 1; \rho, \alpha) \) for given \( l_j(s; \rho, \alpha) \). This system can be solved recursively by induction for each index \( j \).

Finally we derive relations, which will be used in Chapter 5 for the treatment of the degradation operator respectively the convolution operator.

LEMMA 2.9. For \( s \geq 2 \) the following relation holds

\[
(2.33) \quad \sum_{r=1}^{s} l_j(s; \rho, \alpha) = \frac{1}{1 - \rho} \left( l_j(s; \rho, \alpha) - l_{j+1}(s; \rho, \alpha) + \rho^{j+1} \binom{j + \alpha}{j + 1} \right) .
\]

Proof. Induction over \( s \) using

\[
l_j(1; \rho, \alpha) = \rho^j \binom{j + \alpha}{j}
\]

and the well-known property

\[
\binom{x + 1}{j} = \binom{x}{j} + \binom{x}{j - 1} , \quad x \in \mathbb{R} , \quad j \in \mathbb{N} .
\]

38
LEMMA 2.10. For $s \geq 1$ we have

\[
(2.34) \sum_{r=1}^{s-1} l_k(r) l_j(s-r) = \frac{1}{1 - \rho} [\rho l_{k+1}(s) - l_{k+j+1}(s)] , \quad l_k(s) = l_k(s; \rho, 0) ,
\]

Proof. Induction over $s$ again.

2.3 APPROXIMATION PROPERTIES

Let $u \in H_{\rho, \alpha}$ be expanded in the orthogonal basis $\{\psi_k(\rho, \alpha)\} = \{\Psi_{\rho, \alpha} l_k(\rho, \alpha)\}$ of $H_{\rho, \alpha}$ by

\[
(2.35) \quad u(s) = \Psi_{\rho, \alpha}(s) \sum_{k=0}^{\infty} a_k l_k(s; \rho, \alpha) ,
\]

where the $\{l_k(\rho, \alpha)\}$ are the modified discrete Laguerre polynomials. The expansion coefficients $a_k$ can be expressed by

\[
(2.36) \quad a_k = \frac{1}{\gamma_k^{\rho, \alpha}} \sum_{s=1}^{\infty} u(s) l_k(s; \rho, \alpha) ,
\]

and it is well known by the Parseval equality that

\[
(2.37) \quad \|u\|_{\rho, \alpha}^2 = \sum_{k=0}^{\infty} a_k^2 \gamma_k^{\rho, \alpha} .
\]

The orthogonal projection to the $n$-dimensional subspace

\[
(2.38) \quad H^n_{\rho, \alpha} = \text{span} \{\psi_0(\rho, \alpha), \ldots, \psi_n(\rho, \alpha)\} \subset H_{\rho, \alpha}
\]

is defined by

\[
(2.39) \quad \mathcal{P}^n_{\rho, \alpha} u(s) := \sum_{k=0}^{n} \frac{1}{\gamma_k^{\rho, \alpha}} (u, \psi_k(s; \rho, \alpha))_{\rho, \alpha} \psi_k(s; \rho, \alpha) = \Psi_{\rho, \alpha}(s) \sum_{k=0}^{n} a_k l_k(s; \rho, \alpha) .
\]

The projection error is easily seen to be

\[
(2.40) \quad \mathcal{Q}^n_{\rho, \alpha} u(s) := u(s) - \mathcal{P}^n_{\rho, \alpha} u(s) = \Psi_{\rho, \alpha}(s) \sum_{k=n+1}^{\infty} a_k l_k(s; \rho, \alpha) .
\]

Obviously we have $\|u - \mathcal{P}^n_{\rho, \alpha} u\|_{\rho, \alpha} \to 0$ as $n \to \infty$ for all $u \in H_{\rho, \alpha}$ and $\|\mathcal{P}^n_{\rho, \alpha}\|_{\rho, \alpha} \leq 1$. We want to estimate the norm of the projection error $\|\mathcal{Q}^n_{\rho, \alpha} u\|_{\rho, \alpha}$.
in terms of higher differences of \( u \) (analogue to higher derivatives in the continuous case), which should be a measure of the smoothness of \( u \) in \( H_{p,\alpha} \). In \( H^{p,\alpha} \), which is spanned by polynomials, we could use the standard forward difference operator, since for \( p \) a polynomial of degree \( m \)

\[
\Delta^{m+1} p(s) = 0 .
\]

However, in \( H_{p,\alpha} \) the basis consists of polynomials multiplied by the weight function \( \Psi_{p,\alpha}(s) \), and it is easily seen that even

\[
\Delta^m \Psi_{p,\alpha}(s) = \Psi_{p,\alpha}(s) \left[ \frac{p(s+\alpha)}{s} - 1 \right]^m \neq 0 \text{ for } m \geq 1 .
\]

This consideration together with (2.23) inspires the following definition of a modified difference operator.

**Definition 2.11.** Let for \( u \in H_{p,\alpha} \) the weighted difference operator \( \Delta_{\alpha} \) be defined by

\[
(2.41) \quad \Delta_{\alpha} u(s) := \Psi_{p,\alpha+1}(s) \Delta \left( \frac{u(s)}{\Psi_{p,\alpha}} \right), \quad \Delta_{\alpha} u \in H_{p,\alpha+1} ,
\]

by means of the forward difference operator \( \Delta \). Higher weighted differences are inductively given by

\[
(2.42) \quad \Delta^m_{\alpha} u(s) := \Delta_{\alpha+m-1} \Delta_{\alpha+m-2} \ldots \Delta_{\alpha} u(s) .
\]

The re-multiplication with \( \Psi_{p,\alpha+1} \) instead of \( \Psi_{p,\alpha} \) has been done for analytical convenience as we will see below. From Lemma 1.2 we know, that if \( u \in H_{p-\varepsilon,\alpha} \), the weighted difference is an element of \( H_{p,\alpha} \).

**Corollary 2.12.** For \( u \in H_{p,\alpha} \) with a representation (2.35) the \( m \)-th weighted higher difference can be written as

\[
(2.43) \quad \Delta^m_{\alpha} u(s) = \Psi_{p,\alpha+m}(s) \sum_{k=0}^{\infty} a_{k+m} (\rho - 1)^m l_k(s; \rho, \alpha + m) .
\]

**Proof.** Again we use the difference relation (2.23)

\[
\Delta l_k(s; \rho, \alpha) = (\rho - 1) l_{k-1}(s; \rho, \alpha + 1) .
\]

Inserting (2.35) we get

\[
\Delta_{\alpha} u(s) = \Psi_{p,\alpha+1} \sum_{k=0}^{\infty} a_k \Delta l_k(s; \rho, \alpha)
\]

\[
= \Psi_{p,\alpha+1} \sum_{k=0}^{\infty} a_{k+1} (\rho - 1) l_k(s; \rho, \alpha + 1) ,
\]

40
which is an expansion of $\Delta_\alpha u$ in $H_{p,\alpha+1}$. Now induction leads to (2.43).

As a consequence of Corollary 2.12 we can easily compute $\|\Delta_\alpha^m u\|_{p,\alpha+m}$. Using the orthogonality of the polynomials $\{l_k(\rho, \alpha)\}$ we obtain

$$ (2.44) \quad \|\Delta_\alpha^m u\|_{p,\alpha+m}^2 = \sum_{k=m}^{\infty} a_k^2 (1-\rho)^{2m} \gamma_{k-m}^{\rho,\alpha+m} < \infty. $$

The following definition prepares the proof of an approximation result.

**Definition 2.13.** For $u \in H_{p,\alpha}$, $m \geq 1$, define the $m$-th weighted difference norm $\|u\|_{p,\alpha,m}$ by

$$ (2.45) \quad \|u\|_{p,\alpha,m}^2 : = \|u\|_{p,\alpha}^2 + \|\Delta_\alpha^m u\|_{p,\alpha+m}^2. $$

For $m = 0$ set

$$ \|u\|_{p,\alpha,0} : = \|u\|_{p,\alpha}. $$

**Remark.** This norms can be regarded as discrete weighted Sobolev norms.

**Theorem 2.14.** For $u \in H_{p,\alpha}$ and $n+1 \geq m \geq 1$ the approximation error $Q_{n,\alpha}^u$ can be estimated by

$$ (2.46) \quad \|Q_{n,\alpha}^u\|_{p,\alpha}^2 \leq \frac{\rho^m}{(1-\rho)^{2m}} \frac{(1+\alpha)(2+\alpha)\ldots (m+\alpha)}{(n+1)n\ldots (n+1-(m-1))} \|u\|_{p,\alpha,m}^2. $$

**Proof.** Inserting (2.44) we start with

$$ (2.47) \quad \|u\|_{p,\alpha,m}^2 = \sum_{k=0}^{\infty} a_k^2 \gamma_k^{\rho,\alpha} + \sum_{k=m}^{\infty} a_k^2 (1-\rho)^{2m} \gamma_{k-m}^{\rho,\alpha+m}, $$

using

$$ \gamma_{k-m}^{\rho,\alpha+m} = \rho^k \gamma_{k-m}^{\rho,\alpha} = \rho (k+\alpha) (k-\alpha) \ldots (k-m+1)(k-m-1) \frac{(1+\alpha)\ldots (m+\alpha)}{(1+\alpha)\ldots (m+\alpha)}. $$

The norm of the projection error can be written as

$$ \|Q_{n,\alpha}^u\|_{p,\alpha}^2 = \sum_{k=n+1}^{\infty} a_k^2 \gamma_k^{\rho,\alpha}. $$
and it can be seen that

\[
1 \leq \frac{\rho^m}{(1 - \rho)^{2m}} \frac{(1 + \alpha)(2 + \alpha) \ldots (m + \alpha)}{(n + 1)n \ldots (n + 1 - (m - 1))} \times \frac{1 + \frac{(1 - \rho)^{2m}}{(1 + \alpha) \ldots (m + \alpha)}}{k(k - 1)\ldots(k - m + 1)}
\]

for \( n + 1 \leq k \). Combining (2.47) and (2.48) gives the assertion (2.46).

We observe that the approximation behavior depends strongly on the constants which are independent of \( n \). For numerical error estimations, we may assume the projection error

\[
\bar{e}_n = \left\| Q_n^{\alpha} u - \| u \|_{P, \alpha} \right\|
\]

for a constant \( C(\rho, \alpha, m) \) and a power \( r \).

It has been turned out in the previous papers [18], [9] and [52], that an error estimation using the leading term of \( Q_n^{\alpha} u \) works well in the present context. This estimate \( \bar{e}_n \) of \( e_n \) can be obtained by

\[
\bar{e}_n^2 = a_{n+1}^2 \gamma_{n+1}^{\rho, \alpha} \quad \text{(see Lemma 3.3)}
\]

If the computation of the coefficients \( a_k \) depends on the truncation index \( n \), this error estimation will be modified as suggested in [18]. We will come back to this point in Section 3.1.
3 Discrete Galerkin Method

Assume \( \varphi \in H_{p,\sigma} \). It can be seen from (1.54), (1.55) that the realization of implicit or explicit Euler steps consists of three tasks:

(i) projections of the form \( \mathcal{P}_{n}^{p,\alpha} \varphi \), \( \mathcal{P}_{n}^{p,\alpha} \) from (2.39).

(ii) projections of the form \( \mathcal{P}_{n}^{p,\alpha} A \varphi \), \( A \) bounded linear operator (explicit Euler step).

(iii) approximate solutions of the countable system of algebraic equations

\[
(I - \tau A)u = \varphi \quad \text{in} \quad H_{p,\sigma}^{n} \quad \text{(implicit or semi-implicit Euler step)}.
\]

As we are interested in Galerkin approximations in \( H_{p,\sigma} \), we assume that \( \varphi, A \varphi \) and the solution \( u \) of the CAE in (iii) (all called \( u \) from now on) have a basis expansion (2.35).

The discrete Galerkin method yields perturbed approximations \( \tilde{u}^{n} \in H_{p,\sigma}^{n} \) of \( u \) of the form

\[
\tilde{u}^{n} = \sum_{k=0}^{n} \tilde{a}_{k} \psi_{k}(\rho, \alpha),
\]

with a truncation index \( n \) and coefficients \( \tilde{a}_{k} \), \( k = 0, 1, \ldots, n \), which may be the exact expansion coefficients \( a_{k} \) or approximations of them. The error \( \| \tilde{u}^{n} - u \|_{p,\sigma} \) of approximation (3.1) can be written in terms of the projection error \( \| u^{n} - u \|_{p,\sigma} \) (compare (2.40)) and a so-called truncation error \( \| \tilde{u}^{n} - u^{n} \|_{p,\sigma} \):

\[
\| \tilde{u}^{n} - u \|_{p,\sigma}^{2} = \| u^{n} - u \|_{p,\sigma}^{2} + \| \tilde{u}^{n} - u^{n} \|_{p,\sigma}^{2} = \sum_{k=n+1}^{\infty} a_{k}^{2} \gamma_{k}^{p,\sigma} + \sum_{k=0}^{n} (\tilde{a}_{k} - a_{k})^{2} \gamma_{k}^{p,\sigma}.
\]

An estimation of error (3.2) and a matching of projection error and truncation error are suggested at the end of the following section.

3.1 Projections and Approximate Solutions

The projection of an element \( u \in H_{p,\sigma} \) to the subspace \( H_{p,\sigma}^{n} \) can formally be written as

\[
u^{n}(s) = \sum_{k=0}^{n} a_{k} \psi_{k}(s; \rho, \alpha) = \Psi_{p,\sigma}(s) \sum_{k=0}^{n} a_{k} l_{k}(s; \rho, \alpha).
\]

In general the expansion coefficients \( a_{k} \) can formally be expressed by relation (2.36). Sometimes properties of the modified discrete Laguerre polynomials \( l_{k}(\rho, \alpha) \) make an analytic calculation of the \( a_{k} \) possible.
Example 3.1. Let

\[ u_s = \frac{s}{r} \hat{p}^s, \quad \hat{p} := e^{-1/r}, \quad s \geq 1, \quad r \text{ given}. \]

This is a realistic setting of initial values in some applications. It is easily seen that

\[ u = \frac{\hat{p}}{(1 - \hat{p})^2} \Psi_{\hat{p},1} \in H_{\hat{p},1}, \]

but it will turn out in Example 5.3, that an approximation in \( H_{\hat{p},1} \) is not feasible in that context. Instead of this, \( u \) can be expanded for \( \rho = 2\hat{p}/(1 + \hat{p}) \) and \( \alpha = 0 \) in the respective space with coefficients

\[ a_k = \frac{\hat{p}}{(1 - \hat{p})^2} 2^{-k}(1 - k), \quad k \geq 0. \]

This can be seen from

\[ s = \frac{1}{\hat{p} - 1} \left( l_1(s; \hat{p}, 0) - l_0(s; \rho, \alpha) \right) \]

and the transformation formula (2.24) in Lemma 2.2.

Whenever there are no analytical properties to obtain the exact coefficients \( a_k \), we have to evaluate the scalar products in (2.36) numerically. This will be done by the summation algorithm described in Chapter 6 and leads to approximations \( \tilde{a}_k \) of \( a_k \) and hence to an expansion (3.1).

Next we consider projections of \( Au \), \( A \) a linear operator. We will try to apply properties of the \( H_{\rho,\alpha} \)-basis as far as possible again and propose a classification of linear CODE's formulated in terms of the projections \( \mathcal{P}_{\rho,\alpha}^n \):

**Definition 3.1.**

(i) \( A \) is called invariant, if

\[ \mathcal{P}_{\rho,\alpha}^n A = \mathcal{P}_{\rho,\alpha}^n A \mathcal{P}_{\rho,\alpha}^n \text{ for all } n \in \mathbb{N}. \]

(ii) \( A \) is called \( m \)-invariant, if there is an \( m > 0 \), such that

\[ \mathcal{P}_{\rho,\alpha}^n A = \mathcal{P}_{\rho,\alpha}^n A \mathcal{P}_{\rho,\alpha}^{n+m} \text{ for all } n \in \mathbb{N}. \]

(iii) otherwise \( A \) is called general.
In literature problems with invariant operators are sometimes called *self-closing*, problems with $m$-invariant or general operators are called *open*. In order to explain Definition 3.1, we consider

$$u \in H_{\rho,\alpha}, \quad u = x + y, \quad x = \mathcal{P}_n^{\rho,\alpha} u, \quad y = (I - \mathcal{P}_n^{\rho,\alpha})u.$$ 

From (3.5) it follows for an invariant operator $A$ that

$$\mathcal{P}_n^{\rho,\alpha} Ay = 0,$$

that means the *complement of $H_{\rho,\alpha}^n$ is invariant* under $A$. Similar considerations turn out that with (3.6) an $m$-invariant operator $A$ maps the complement of $H_{\rho,\alpha}^m$ onto the complement of $H_{\rho,\alpha}^n$. By this notions the backward shift operator $S_-$ is invariant, since

$$\mathcal{P}_n^{\rho,\alpha} S_- u = \sum_{j=0}^{n} \frac{1}{\gamma_j^{\rho,\alpha}} (S_- u, \psi_j)_{\rho,\alpha} \psi_j = \sum_{j=0}^{n} \frac{1}{\gamma_j^{\rho,\alpha}} (u, S_+ \psi_j)_{\rho,\alpha} \psi_j.$$ 

With relation (2.31) the last scalar product can be expressed in terms of the coefficients $a_k$, $k \leq j$, if we assume an $H_{\rho,\alpha}$-expansion of $u$. For the forward shift operator $S_+$ things are different:

$$\mathcal{P}_n^{\rho,\alpha} S_+ u = \sum_{j=0}^{n} \frac{1}{\gamma_j^{\rho,\alpha}} \left( (u, S_- \psi_j)_{\rho,\alpha} - u_1 \psi_j(0) \right) \psi_j,$$

where the component $u_1$ depends on all expansion coefficients $a_k$ of $u$. Thus $S_+$ is a general operator and this makes the classification from the beginning of Section 1.3 consistent with the above definition in $H_{\rho,\alpha}$: if $(Au)_r$ depends on the component $u_{s+r}$, $r \in \mathbb{N}$, it can be expressed in terms of $S_+$.

As a last example we mention that the degradation operator is 1-invariant (see Example 5.3).

For the treatment of $m$-invariant or general operators we have to replace

$$\mathcal{P}_n^{\rho,\alpha} A \rightarrow \mathcal{P}_n^{\rho,\alpha} A \mathcal{P}_n^{\rho,\alpha}$$

for some $n > 0$. That means we use the coefficients $a_0, \ldots, a_{n+\hat{n}}$ of $u$ to obtain approximations of the first $n+1$ expansion coefficients of $Au$. Obviously we expect, that these approximations will become better for increasing $\hat{n}$.

As in the situation of projections of elements above, the scalar products

$$(Au, \psi_j)_{\rho,\alpha} \text{ respectively } (A\psi_k, \psi_j)_{\rho,\alpha}, \quad j, k = 0, 1, \ldots, n,$$

can only be evaluated by numerical approximation in general, leading to a perturbed expansion (3.1).
The application of the implicit Euler scheme to an abstract ODE (1.52) with step size \( \tau \) requires the solution of systems

\[
(I - \tau A) u = \varphi, \quad A \text{ linear,}
\]

with respect to \( u \). For nonlinear operators the semi-implicit Euler method \([16]\) (see Section 5, (4.1)) can be used, leading to problems of the same structure.

We assume that for \( \varphi \in H_{p,\alpha} \) (3.8) has a unique solution \( u \) in \( H_{p,\alpha} \). With the orthogonal projection (2.39) the Galerkin equations

\[
u^n - \tau \mathcal{P}_n^{p,\alpha} A u^n = \mathcal{P}_n^{p,\alpha} \varphi, \quad u^n \in H_{p,\alpha}^n, \quad n = 0, 1, \ldots
\]

are equivalent to

\[
u^n - \tau \mathcal{P}_n^{p,\alpha} A u^n = \mathcal{P}_n^{p,\alpha} \varphi, \quad u^n \in H_{p,\alpha}^n, \quad k = 0, \ldots, n.
\]

Inserting the basis expansion of \( u \) in \( H_{p,\alpha} \), for fixed \( n \) this is seen to be a linear system:

\[
(I - \tau B) a = b, \quad B := (b_{jk}) := (A \psi_k, \psi_j)_{p,\alpha},
\]

\( a = (a_0, \ldots, a_n) \) the coefficients of \( u \) and \( b = (b_0, \ldots, b_n) \) the coefficients of \( \varphi \) in the \( H_{p,\alpha} \) - basis. The assembling of these equations requires projections of the form \( \mathcal{A}u \), respectively \( \mathcal{A} \psi_j(p,\alpha) \), \( j = 0, \ldots, n \), implying the application either of analytical properties or of numerical summation. For \( m \)-invariant or general operators, the Galerkin equations depend on the truncation index \( \bar{n} \) from (3.7) or the summation tolerance. Thus the Galerkin approximation will be of the form (3.1) again. In this work we will only treat two cases theoretically and omit an examination of perturbed Galerkin equations:

(i) \( A \) is the generator of a \( C_0 \)-semigroup of contractions

(ii) \( \tau A \) is contractive, i.e. \( \tau \|A\|_{p,\alpha} \leq 1 \)

We ask for existence of solutions \( u^n \) of the Galerkin equations (3.9) and for estimates \( \|u - u^n\|_{p,\alpha} \). The following theorem, essentially taken from \([53]\), Theorem 21.G, is a standard result for Galerkin methods in Hilbert spaces adapted to our case.

**Theorem 3.2.** Let \( A : D(A) \subset H_{p,\alpha} \rightarrow H_{p,\alpha} \) be a linear operator. For the following cases the problem (3.8) has a unique solution in \( H_{p,\alpha} \) and the Galerkin method converges, i.e. \( \|u^n - u\| \rightarrow 0 \) for \( n \rightarrow \infty \).
(i) $\tau A$ is contractive. Then the estimation

\[ ||u - u^n|| \leq (1 - \tau ||A||_{\rho,\alpha})^{-1} ||u - P_n^{\rho,\alpha}u|| \]

holds.

(ii) $A$ is generator of a $C_0$-semigroup of contractions and additionally invariant. Then we have convergence for $\tau > 0$ and for the Galerkin solution $u^n$ holds

\[ u^n = P_n^{\rho,\alpha}u. \]

Proof. (i) As $||P_n^{\rho,\alpha}A||_{\rho,\alpha} \leq ||A||_{\rho,\alpha}$ the equations (3.8) and (3.9) have unique solutions by the Banach fixed-point theorem. Further

\[ ||(I - \tau P_n^{\rho,\alpha}A)^{-1}||_{\rho,\alpha} \leq (1 - \tau ||A||_{\rho,\alpha})^{-1} \]

by use of the Neumann series. Summarizing (3.8) and (3.9) we obtain

\[ (I - \tau P_n^{\rho,\alpha}A)(u - u^n) = u - P_n^{\rho,\alpha}u \]

and therefore (3.12) and the convergence for $n \to \infty$.

(ii) We know that $(I - \tau A)$ is invertible for all $\tau > 0$. The question is, what can be said about $(I - \tau P_n^{\rho,\alpha}A)^{-1}$? From perturbation theory it is known, that the uniform boundedness of

\[ ||A - P_n^{\rho,\alpha}A||_{\rho,\alpha} \quad \text{for} \quad n \to \infty \]

is crucial (see e.g. [34], IV.3, IV.4). This condition is met for compact operators $A$ and the convergence of the Galerkin method can be proven ([53], Theorem 21.G,(b)). For $A$ invariant a result by means of semigroup theory can be derived. The infinitesimal generator $A$ of a $C_0$-semigroup of contractions is characterized by the Lumer-Phillips theorem:

$A$ is dissipative, i.e. $(Au, u) \leq 0$ and for a $\lambda_0 > 0$ the range of $\lambda_0 I - A$ is $H_{\rho,\alpha}$. For $u^n \in H^n_{\rho,\alpha}$ it follows from

\[ (P_n^{\rho,\alpha}Au^n, u^n)_{\rho,\alpha} = (Au^n, u^n)_{\rho,\alpha} \]

that $P_n^{\rho,\alpha}A$ is also dissipative and the range condition is fulfilled in $H^n_{\rho,\alpha}$ since $P_n^{\rho,\alpha}A = P_n^{\rho,\alpha}AP_n^{\rho,\alpha}$. Hence $P_n^{\rho,\alpha}A$ generates a contractive $C_0$-semigroup too. Note that if the original and the Galerkin equation are uniquely solvable and $A$ is invariant, we always have $u^n = P_n^{\rho,\alpha}u$. 

The derivation of the Galerkin equations uses (if possible) properties of the discrete Laguerre polynomials. As an illustration we consider again the
Example 3.2: Backward difference operator. We evaluate

\[ (-\nabla u, \psi_j)_{\rho,\alpha} = - \sum_{s=1}^{\infty} u_s \Delta_j(s) . \]

Then

\[ (-\nabla u, \psi_j)_{\rho,\alpha} = \sum_{k=0}^{\infty} a_k \sum_{\nu=0}^{j-1} (\rho - 1)^{\nu} (l_{\nu}, l_j)_{\rho,\alpha} \]

\[ = \frac{1 - \rho}{\rho(j+\alpha)} \gamma_{j+\alpha}^{\rho,\alpha} \sum_{k=0}^{j-1} a_k \binom{k+\alpha}{k} \]

by use of (2.31), the orthogonality of the \( l_j \) and the definition of \( \gamma_{j+\alpha}^{\rho,\alpha} \). With \( \varphi = \sum_{k=0}^{\infty} b_k \psi_k \) the Galerkin equations for \((I + \tau \nabla)u = \varphi\) are

\[ a_j + \tau \frac{1 - \rho}{\rho(j+\alpha)} \sum_{k=0}^{j-1} a_k \binom{k+\alpha}{k} = b_j, \quad j = 0, 1, \ldots, n , \]

and can be solved recursively in this case. Generally, Gauss elimination has to be applied.

Error estimation. The numerical implementation of the discrete Galerkin method requires error estimations of the projection and the truncation error. A computable (and cheap) estimation of the projection error is available.

**Lemma 3.3.** Define an error estimation \( \varepsilon_n \) by

\[ \varepsilon_n^2 := ||u^{n+1} - u^n||^2 = a_{n+1}^2 \gamma_{n+1}^{\rho,\alpha} , \]

and assume that there exist \( C < 1 \) and \( n_0 \geq 1 \) such that for \( n > n_0 \) the relation

\[ \varepsilon_{n+1} \leq C \varepsilon_n \]

holds. Then

\[ \varepsilon_n \leq \varepsilon_n \leq \left( \frac{1}{1 - C^2} \right)^{1/2} \varepsilon_n, \quad n \geq n_0 . \]

**Proof.** Obviously

\[ \varepsilon_n^2 \leq \varepsilon_n^2 = \sum_{k=n+1}^{\infty} a_k^2 \gamma_k^{\rho,\alpha} \]
since $\gamma_k^{p,\alpha} > 0$. On the other hand it follows from (3.17) for $n \geq n_0$ that

$$
\varepsilon_n^2 = \sum_{k=n+1}^{\infty} \varepsilon_{k-1}^2 \leq \varepsilon_n^2 \sum_{k=0}^{\infty} C^{2k} = \frac{1}{1-C^2} .
$$

The underestimation of the true error depending on $C$ can be numerically controlled by setting a safety factor. In applications the projection error has to be measured in a scaled norm, of course.

**Example 3.3.** It is easily seen that in Example 3.1, \(u\) approximated in $H_{p,1}$,

$$
\frac{\varepsilon_{n+1}}{\varepsilon_n} \leq \frac{3}{5} , \quad \text{for} \quad n \geq 6 ,
$$

hence the underestimation of the error is smaller than $5/4$. A comparison between true and estimated error for this example can be found in [18], Table 1.

Now assume, that there is an algorithm producing successive approximations $a_k^n$ of $a_k$ -- e.g by setting $m = \bar{n} = n, n+1, \ldots$, in (3.7) or by a refinement strategy during the numerical summation with $m$ the refinement level or the index of a tolerance. Then the following error estimation will be applied

$$
(3.19) \quad \varepsilon_{n,m}^2 = \sum_{k=0}^{n} (a_k^m - a_k^{m+1})^2 \gamma_k^{p,\alpha} + (a_{n+1}^m)^2 \gamma_{n+1}^{p,\alpha} .
$$

We are only interested in a truncation error (respectively its estimation) $\varepsilon_{T,n,m}$ being just a little smaller than the projection error $\varepsilon_{P,n,m}$. To be more precise, we require that

$$
(3.20) \quad \|u_{n+1,m(n+1)} - u_n\|_{p,\alpha} < \kappa \|u_{n,m(n)} - u\|_{p,\alpha}
$$

with some safety factor $0 < \kappa < 1$ (see also [17], (1.26)). In actual computations, the terms on both sides are replaced by the estimates suggested in (3.19), $\kappa$ is set to $1/4$.

### 3.2 Weight Function Fitting

In order to minimize the computational effort of the discrete Galerkin method, a good choice of the parameters $\rho$ and $\alpha$ is crucial. Strictly speaking, for $u \in H_{p,\alpha}$ and $\varepsilon > 0$ it would be optimal to find $\bar{\rho}, \bar{\alpha}$ such that $u \in H_{\bar{p},\bar{\alpha}}$ and

$$
(3.21) \quad \|u - u^n\|_{p,\alpha} < \varepsilon , \quad u^n = P_{p,\alpha}u ,
$$

for $n$ as small as possible. However, even if the error $\|u - u^n\|$ is computable, the estimation of $\bar{\rho}, \bar{\alpha}$ turns out to be a nonlinear minimization problem, since
transformations between $H_{\rho,\alpha}$ - spaces have to be performed. The interesting question: 'What criterions ensure that $u$ is nearer (in the sense of orthogonal projection in Hilbert spaces) to $H_{\rho,\alpha}$ than to $H^a$ in general?' is left to further examinations.

In this work, we choose an heuristic approach which is an extension of the moving weight function concept suggested in [18]. The main idea is the fitting of the first statistical moments of $u \in H_{\rho,\alpha}$ and the weight function $\Psi_{\rho,\alpha}$, a procedure which is possible by Corollary 1.5. Due to the normalization of the family $\Psi_{\rho,\alpha}$ this leads to an implicit definition of the parameters $\bar{\rho}, \bar{\alpha}$ by

\[
\frac{\mu_1[u]}{\mu_0[u]} = \mu_1[\Psi_{\rho,\alpha}] = \nu_1(\bar{\rho}, \bar{\alpha}),
\]

\[
\frac{\mu_2[u]}{\mu_0[u]} = \mu_2[\Psi_{\rho,\alpha}] = \nu_2(\bar{\rho}, \bar{\alpha}),
\]

using the notations (2.2), (2.3). From (3.22), (2.2), (2.3) we can deduce by some technical manipulations ($\mu_k = \mu_k[u]$):

\[
\bar{\rho}(u) = \bar{\rho} = \frac{\mu_0 \mu_2 - \mu_2^2 - \mu_1 \mu_0 + \mu_0^2}{\mu_0 \mu_2 - \mu_2^2},
\]

\[
\bar{\alpha}(u) = \bar{\alpha} = \frac{2 \mu_2^2 - \mu_1 \mu_0 - \mu_2 \mu_0}{\mu_0 \mu_2 - \mu_1 \mu_0 + \mu_0^2}
\]

The requirements $0 < \bar{\rho} < 1$ and $\bar{\alpha} > -1$ are fulfilled, if the denominators of the expressions in (3.23) are positive. Whenever only the parameter $\rho$ has to be adapted (i.e. $\alpha = 0$), (3.22) reduces to (consistent with [18], (3.15))

\[
\bar{\rho}(u) = 1 - \frac{\mu_0[u]}{\mu_1[u]}.
\]

In order to compute the moments $\mu_0[u], \mu_1[u], \mu_2[u]$ for $u \in H_{\rho,\alpha}$ given in the basis expansion, we use the fact, that the monomials $s^k$ can be represented in terms of the polynomials $l_k(\rho, \alpha)$ by

\[
s^k = \sum_{m=0}^{k} b_{km} l_m(s), \quad k = 0, 1, \ldots,
\]

with coefficients $b_{km} = b_{km}(\rho, \alpha)$, $b_{kk} \neq 0$. Then $\mu_k[\Psi_{\rho,\alpha}] = b_0$ and insertion of $u$ into the definition of statistical moments yields

\[
\mu_k[u] = \sum_{m=0}^{k} b_{km} a_m \gamma_{m}^{\rho,\alpha}.
\]
The coefficients $b_{km}$ can be computed here to be

$$b_{10} = \nu_1(\rho, \alpha), \quad b_{11} = \frac{1}{1 - \rho}$$

and

$$b_{20} = \nu_2(\rho, \alpha), \quad b_{21} = \frac{2\alpha \rho + \rho + 3}{(1 - \rho)^2}, \quad b_{22} = \frac{2}{(1 - \rho)^2}.$$ 

In general, the condition (3.22) has a nice characterization for weight functions $\Psi$ with $p$ parameters and associated orthogonal polynomials. Assume that the statistical moments $\nu_k$ of $\Psi$ are known and $\nu_0 = \gamma_0 = 1$. By (3.26) the condition (3.22) is equivalent to

$$\mu_0[\gamma] = a_0,$$

(3.27)

$$\sum_{m=0}^{k} b_{km} a_m \gamma_m = b_{k0} a_0, \quad 1 \leq k \leq p.$$ 

From (3.27) we obtain by induction

(3.28)

$$a_1 = a_2 = \ldots = a_p = 0.$$ 

This property has been used in [18] for $p = 1$ to derive a differential equation for the parameter of the Galerkin method.

**Remark.** In order to ensure a reliable computation of $\bar{\rho}$, $\bar{\alpha}$ for (possibly) perturbed coefficients $\bar{a}_0, \bar{a}_1, \bar{a}_2$, additionally to condition (3.20) the following relations should be fulfilled

(3.29)

$$|\bar{\rho}(\bar{u}^n) - \bar{\rho}(u^n)| \leq \varepsilon_{\rho},$$

$$|\bar{\alpha}(\bar{u}^n) - \bar{\alpha}(u^n)| \leq \varepsilon_{\alpha},$$

with $\bar{u}^n$ from (3.1) and appropriate (relative) accuracies $\varepsilon_{\rho}, \varepsilon_{\alpha}$. Obviously the left-hand sides in (3.29) are replaced by estimates in the actual implementation.

**Connection with Charlier polynomials.** The weight function fitting condition leads us in a natural way to the *Charlier polynomials* associated to the Poisson distribution.

Let

$$u(s; \lambda) = e^{-\lambda} \frac{\lambda^{s-1}}{(s-1)!}, \quad s \in \mathbb{N},$$

a Poisson distribution. The moments $\mu_0, \mu_1, \mu_2$ of $u$ are

$$\mu_0 = 1, \quad \mu_1 = \lambda + 1, \quad \mu_2 = \lambda^2 + 3\lambda + 1.$$
Insertion of these moments into (3.23) unfortunately leads to $\bar{\rho} = 0$ and $\bar{\alpha}$ infinite. But it turns out that formally $\bar{\rho} \bar{\alpha} = \lambda$. This suggests the examination of $\Psi_{\rho, \lambda/\rho}$ for $\rho \to 0$. A short calculation yields

$$\lim_{\rho \to 0} \Psi_{\rho, \lambda/\rho}(s) = u(s; \lambda).$$

Moreover, if we insert the parameters $\rho, \alpha = \lambda/\rho$ into the series representation (2.21) of the modified discrete Laguerre polynomials we obtain

$$\lim_{\rho \to 0} l_n(s; \rho, \lambda/\rho) = \frac{\lambda^{-n}}{n!} \sum_{k=0}^{n} \binom{n}{k} k! (-1)^k \lambda^{-k} \binom{s-1}{k} = \frac{\lambda^{-n}}{n!} c_n(s; \lambda),$$

with $c_n(s; \lambda)$ the Charlier polynomials due to the representation [18], (3.24). This is a very nice aspect and shows that we have closed the gap between the different discrete Galerkin methods used up to now.

Remark. The possibility of a transition from modified discrete Laguerre (Meixner) polynomials to Charlier polynomials is not a new finding, of course, but part of the important Askey scheme [3], [35]. The Askey scheme summarizes those polynomials, which are called classical orthogonal polynomials today. As one characteristic of that scheme, the number of parameters of a polynomial set is reduced by one parameter, if a transition to another polynomial set is performed. The idea of this section – setting the parameters by considering the respective statistical moments – may be new aspect.
4 NUMERICAL REALIZATION OF THE ALGORITHM

In this chapter we describe the essential details of the algorithm:

- time-step and order control, extrapolation
- realization of the discrete Galerkin method

4.1 REALIZATION OF THE TIME-STEP CONTROL

In this section, which follows the presentation of Chapter 3 in [8], it will be explained, that the usual results for extrapolation—methods with some modifications still hold in $H_{p,a}$, instead of $I^n$. In the fully-discrete case (i.e. time discretization and approximation in $H_{p,a}$ by means of orthogonal polynomials) the requirements of the original problem have to be taken into account.

We use the implicit (explicit, semi-implicit) Euler discretization in time and a time-step and order control of the method by extrapolation following the ideas of DEUFLHARD [15] for ODE’s and of BORNEMANN [8] for parabolic PDE’s. As pointed out in Section 1.4, the explicit Euler discretization is applied only to bounded operators, the implicit Euler scheme only to linear problems. For the nonlinear case (see Example 5.4) the semi-implicit Euler scheme [16] is applied:

\[
(I - \tau A) \Delta u(t) = \tau f(u(t)), \\
u_{t}(t + \tau) := u(t) + \Delta u(t),
\]

with $A = f_u(u(t))$ the Frechet derivative of the right-hand side $f(u)$ in problem (1.1) at time $t$. Obviously the implicit Euler discretization is identical with the semi-implicit Euler scheme in the linear case. The extrapolation on the basis of one of these schemes works as follows:

The algorithm suggests an outer time step $T > 0$ for which

\[
U_{t1} := u_{t1}(T),
\]

the Euler discretizations with time-step $\tau_t = \frac{T}{n_t}$ as introduced in Section 1.4, are computed for a given sequence of increasing $n_t$:

\[
F = \{n_1, n_2, \ldots\}.
\]

Since $\lim_{\tau \to 0} u_{t}(T) = u(T)$, the values $(U_{t1}, \ldots, U_{t1})$ are extrapolated to $\tau = 0$, getting an hopefully better approximation than the $U_{t1}$. This will be explained now. We compute the interpolation polynomial with values in $H_{p,a}$

\[
p_k(\tau) = e_0 + e_1 \tau + \ldots + e_{k-1} \tau^{k-1},
\]

53
\( e_0, \ldots, e_{k-1} \in H_{p,\alpha} \), such that

\[ p_{jk}(\tau_i) = U_{i1} \text{ for } i = j, j - 1, \ldots, j - k + 1. \]  

(4.5)

This can be done in \( H_{p,\alpha} \), since the \( e_j \) can be determined as linear combinations of the \( U_{i1} \), as we will see later on. Now extrapolation to \( \tau \downarrow 0 \) leads to

\[ U_{jk} := p_{jk}(0) = e_0 \in H_{p,\alpha}. \]  

(4.6)

The values \( U_{jk} \) can easily be computed in the extrapolation table

\[
\begin{array}{c c c c}
U_{i1} & \downarrow & \downarrow & \downarrow \\
\rightarrow & U_{21} & \rightarrow & U_{22} \\
& \vdots & \vdots & \vdots \\
& U_{k1} & \rightarrow & \cdots & U_{k,k-1} & \rightarrow & U_{kk}
\end{array}
\]  

(4.7)

using the Aitken–Neville algorithm: \( j \geq 2 \)

\[ U_{jk} = U_{j,k-1} + \frac{U_{j,k-1} - U_{j-1,k-1}}{n_{j,k-1} - 1}, \quad k = 2, \ldots, j, \]  

(4.8)

which can also be performed in \( H_{p,\alpha} \).

The error \( \|u(T) - U_{jk}\|_{p,\alpha} \) will be examined in the following

**Theorem 4.1.** For \( u_0 \in H_{p,\alpha} \) we have

\[ \epsilon_{jk} := \|u(T) - U_{jk}\|_{p,\alpha} \leq \gamma_{jk} T^{k+1}, \]  

(4.9)

where asymptotically

\[ \gamma_{jk} \doteq [n_{j-k+1} \ldots n_j]^{-1} C, \]  

(4.10)

\( C \) depending on the problem, on the initial value and on \( F \).

**Proof.** Follows as usual from Theorem 1.13. For instance take the proof of Theorem II. 9.1 in [27].

By these remarks and the fact that in general \( \gamma_{jk} \) decreases for increasing \( k \) we see, that the assumption

\[ \epsilon_{j,k+1} \leq c\epsilon_{jk}, \quad c < 1, \]  

(4.11)
is reasonable. As in DEUFLHARD [15] Section 1.2 we are thus led to the subdiagonal error criterion

$$\epsilon_{k+1,k} = \|U_{k+1,k} - U_{k+1,k+1}\| =: [\epsilon_{k+1,k}]_{sd}$$

as a reasonable (semi-discrete) estimator.

**Notation:** A single quantity in square brackets denotes a computable estimator for this quantity.

Summarizing, the basic time-step for achieving a prescribed tolerance TOL in line $j + 1$ of the extrapolation table is given as

$$T_{j+1,j} := \left( \frac{TOL}{[\epsilon_{j+1,j}]_{sd}} \right)^{1/(k+1)} T,$$

$T$ the present basic time-step.

In the fully discrete case we have to approximate the solution of the countable systems of algebraic equations (CAE’s) arising in each (semi-) implicit Euler step, respectively the application of the operator to an element from $H_{\rho,\alpha}$ in the explicit case. This will be done by means of the discrete Galerkin method as suggested in Chapter 3. In both cases we obtain the perturbed extrapolation table

$$U_{11} + \delta_{11} \downarrow \quad \vdots \quad \downarrow \quad U_{k1} + \delta_{k1} \rightarrow \ldots \rightarrow U_{kk} + \delta_{kk},$$

instead of (4.7), where the $\delta_{ij}$ are produced by successive (numerical) solution of the CAE’s (or by numerical projection) and the $\delta_{jk}$, $k > 1$, are the propagated errors in the table. Let $\bar{U}_{jk} := U_{jk} + \delta_{jk}$. The task now is to give a criterion for the accuracy of the discrete sub-problems (CAE, projection), such that the propagation of the introduced errors in the extrapolation table leads only to negligible errors in the semi-discrete problem. Therefore we need

(i) a fully discrete estimator $[\epsilon_{k+1,k}]$ with

$$[\epsilon_{k+1,k}]_{sd} \leq [\epsilon_{k+1,k}]$$

and

(ii) a control of $\delta_{k+1,k+1}$, such that

$\bar{U}_{k+1,k+1}$ is a tolerable approximation.
The estimator \([\epsilon_{k+1,k}]\) will replace \([\epsilon_{k+1,k}]_{sd}\) in the time-step prediction formula (4.13). First we can state

\[
[\epsilon_{k+1,k}]_{sd} \leq ||\bar{u}_{k+1,k} - \bar{u}_{k+1,k+1}||_{\rho,\alpha} + ||\bar{\delta}_{k+1,k} - \delta_{k+1,k+1}||_{\rho,\alpha}.
\]

Under the assumption that we are able to estimate the perturbations

\[
\delta_{k+1,k},\delta_{k+1,k+1},\delta_{k+1,k} - \delta_{k+1,k+1}
\]

we define

\[
[\epsilon_{k+1,k}] := ||\bar{u}_{k+1,k} - \bar{u}_{k+1,k+1}||_{\rho,\alpha} + [\delta_{k+1,k} - \delta_{k+1,k+1}],
\]

which is a completely computable quantity. It is reasonable to ask for

\begin{align*}
\text{a)} & \quad [\delta_{k+1,k}, [\delta_{k+1,k+1}] \leq \text{TOL}/2 \\
\text{b)} & \quad [\epsilon_{k+1,k}] \leq \text{TOL},
\end{align*}

for given local tolerance TOL. By examination of the propagation of the errors \(\delta_{j1}\) in the table we get

\[
\delta_{jk} = \sum_{i=j-k+1}^{j} \beta_{jk}^{i} \delta_{i1},
\]

where the coefficients \(\beta_{jk}^{i}\) only depend on the chosen subdividing sequence \(\mathcal{F}\). Substituting the \(\delta_{jk}\) by estimates

(4.15) \[ [\delta_{jk}] := \sum_{i=j-k+1}^{j} |\beta_{jk}^{i}| \delta_{i1} \]

we need

(4.16) \[ [\delta_{j1}] \leq \alpha_{j}^{k}\text{TOL}, \]

where the coefficients \(\alpha_{j}^{k}\) can be computed once at the beginning, only depending on \(\mathcal{F}\). The coefficients can be optimized if the amount of work for the computation of \(\mathcal{U}_{j1}\) is known. The building of the extrapolation table up to row \(k\) requires – according to (4.16) – the computation of the \(\mathcal{U}_{j1}\) with error not exceeding \(\alpha_{j}^{k}\text{TOL}\). This is done by solving \(j\) CAE's (performing \(j\) projections), the implicit (explicit) Euler steps. The \(i\)-th of them produces its own error \(\Delta_{i}\) and the exact problem propagates the previous error \(\Delta_{i-1}\) by a propagation operator \(\pi\), thus leading to

(4.17) \[ \Delta_{i} = \Delta_{i} + \pi \Delta_{i-1}. \]

The role of \(\pi\) can be controlled in some cases. As the following result is not a specific property of \(H_{\rho,\alpha}\), we assume a Banach space \(X\) with norm \(|| \cdot ||\).
Lemma 4.2. (i) Assume that the explicit Euler scheme (time step \( \tau \)) is applied to a (nonlinear) problem with operator \( f \) having a Lipschitz constant \( M \). For \( \tau M \leq 1 \) we obtain the estimate

\[
\| \pi \Delta \| \leq \| \Delta \| ,
\]

for the propagation of a previous error \( \Delta \), i.e. \( \Delta \) is not propagated.

(ii) For the implicit Euler discretization applied to a linear problem with \( A \) generator of a \( C_0 \)-semi-group of contractions

\[
\| \pi \| \leq 1 .
\]

holds.

Proof. (i) Let \( u^0 = u(t) \), \( u^1 = u(t + \tau) \). If \( u^0 \) is perturbed by an error \( \Delta \), the explicit discretization yields

\[
u^1 + \pi \Delta = u^0 + \tau f(u^0 + \Delta).
\]

As \( u^0 - u^1 = -\tau f(u^0) \) this leads to

\[
\| \pi \Delta \| = \tau \| f(u^0 + \Delta) - f(u^0) \| \leq \tau M \| \Delta \| \leq \| \Delta \| ,
\]

(ii) By a similar procedure as in (i) we obtain for the implicit case

\[
\pi = (I - \tau A)^{-1} = \frac{1}{\tau} R \left( \frac{1}{\tau}; A \right)
\]

and the Hille-Yosida theorem (Theorem 1.11) gives the assertion.

Now assume that there is a reliable error estimation for approximation of the sub-problems of an Euler step. According to (4.17) and (4.18), (4.19) we get

\[
\| \Delta_j \|_{p,j} \leq j \epsilon ,
\]

\( \epsilon \) the tolerance per step. Using (4.16) this results in the condition

\[
\epsilon := \frac{1}{j} \alpha_j^k \text{TOL}
\]

as accuracy of the approximation of Euler steps leading to \( \mathcal{U}_j \) for an extrapolation table up to row \( k \). This is BORNEMANN'S fundamental connection between the time-control mechanism (extrapolation table) and the space discretization. The coefficients \( \alpha_j^k \) are shown in Table 4.1 for \( \mathcal{F} = \{1, 2, 3, \ldots \} \), the harmonic sequence, optimized in [8] with respect to an amount of work assumed to be \( n \approx \sqrt{\epsilon} \), \( n \) the truncation index in the discrete Galerkin method. This assumption, though not verified theoretically, turned out to be the best empirical setting over all examples.
The order control mechanism. As described by Deuflhard [15] we control the 'order', represented here by the row in the extrapolation table, simultaneous with the time-step. Relation (4.13) (replacing the semi discrete by the fully discrete error estimation) supplies us with step-size guesses $T_{j+1,j}$ for convergence of $U_{j+1,j}$, that means the algorithm expects $U_{j+1,j}$ to be near to the solution within the given tolerance. As in [15] we define

\[ W_{j+1,j} := \frac{T}{T_{j+1,j}} A_{j+1} \]  

the normalized work per unit step, where $A_{j+1}$ measures the amount of work for achieving $U_{j+1,j+1}$, which is in turn depending on the work required by the CAE solver aiming at accuracy $\epsilon$ given in (4.21). But this $\epsilon$ does not only depend on $j$, the row of the table, but also on $k$, the final row, to which the table will be built up. Thus we should replace (4.22) by

\[ W_{j+1,j}^k := \frac{T}{T_{j+1,j}} A_{j+1}^k , \]

introducing $A_{j+1}^k$ as the amount of work for obtaining $U_{j+1,j+1}$ in a table up to $U_{kk}$. These $A_{j+1}^k$ will depend on the CAE solver or the projection mechanism. On this basis we can actually determine an optimal column index $q$ by

\[ W_{j+1,q}^{q+1} = \min_{j=1,\ldots,k-1} W_{j+1,j}^{q+1} . \]

Knowing this $q$, we use the step-size guess $T_{q+1,q}$ for the next basic time-step and expect convergence in the vicinity of $q$. In order to get a reliable code, avoiding pseudo-convergence and related undesirable things which occur in practice three devices have to be implemented.

- convergence monitor
- order window
- a device for possible increase of order greater than $q$,

- see for instance Deuflhard [16].
Realization of the Extrapolation. The extrapolation can be performed in two ways, depending on the approximation of the Euler steps.

(i) Assume that the initial value $u(t)$ for a global time step $T$ is given in $H_{\rho,\alpha}$ and that Galerkin approximations of the $\mathcal{U}_{11}, j = 1, \ldots, k,$ are also computed in $H_{\rho,\alpha}$:

$$
\mathcal{U}_{11} = \sum_{i=0}^{n^{11}} a_i^{11} \psi_i(\rho, \alpha),
$$

$$
\vdots
$$

$$
\mathcal{U}_{k1} = \sum_{i=0}^{n^{k1}} a_i^{k1} \psi_i(\rho, \alpha),
$$

with $n^{j+1,1} \geq n^{j,1}$, $1 \leq j \leq k - 1$. Then the extrapolation table can be built up in $H_{\rho,\alpha}$ by applying the Aitken-Neville scheme to the expansion coefficients $a_i^{k1}$ leading to

$$
u^n_T(t + T) := \mathcal{U}_{kk} = \sum_{i=0}^{n} a_i^{kk} \psi_i(\rho, \alpha), \; \tilde{n} = n^{kk}.
$$

Finally $u^n_T(t + T)$ is transformed to a space $H_{\rho,\alpha}$ due to the weight function fitting condition (3.23) derived in Section 3.2. The procedure above has the advantage that there are no transformations necessary between $H_{\rho,\alpha}$-spaces necessary during the global time step. Furthermore we know from the theory that extrapolation in the basic space $H_{\rho,\alpha}$ is possible, whereas changing the space for each Euler step can lead to non-feasible transformations. This reliability is paid by requiring possibly more expansion coefficients for the single Euler steps, since the the space might be not 'optimal', i.e. not chosen due to the fitting condition (3.22).

(ii) In a second approach the $\mathcal{U}_{ij}$ are computed in spaces chosen by a weight function fitting in each step:

$$
\mathcal{U}_{11} = \sum_{i=0}^{n^{11}} a_i^{11} \psi_i(\rho^{11}, \alpha^{11}),
$$

$$
\vdots
$$

$$
\mathcal{U}_{k1} = \sum_{i=0}^{n^{k1}} a_i^{k1} \psi_i(\rho^{k1}, \alpha^{k1}).
$$

In order to extrapolate the coefficients, now we have to transform the expansion of $\mathcal{U}_{11}$ in $H_{\rho^{11},\alpha^{11}}$ to the space $H_{\rho^{j+1,1},\alpha^{j+1,1}}$ first. As the table is built up row by row, the final space is not known a priori, hence up to $k - 1$ transformations are

59
necessary. This implies several disadvantages: The extrapolation control may be disturbed, since the transformations roughen the entries of the table. The norms used in the error estimation are changed within the extrapolation table. Moreover, for problems with \( m \)-invariant or general operators the parameters \( \rho, \alpha \) possibly depend on the truncation index, such that additional devices have to be implemented. The most important point is, that the concept of the controlled perturbation of the extrapolation table described in Section 4.1 is destroyed. Finally, in case (i) the scalar products used for the assembling of the Galerkin equations have only to be computed once in one global time step, whereas in the second approach it may happen, that even an increase of the truncation index makes a new computation of all scalar products necessary.

Owing to the above considerations the first way seems to be more promising, extended by an additional feature which is called \textit{coefficient reduction}. After transformation of \( u^{(t-f)} \) to a space \( H_{k,\alpha'} \), we choose a (minimal) truncation index \( \tilde{n} \) such that \( \varepsilon_n \) will be smaller than the prescribed local tolerance. This avoids instabilities for higher coefficients introduced by the transformation and has been successfully applied in all considered examples.

4.2 \textbf{REALIZATION OF THE GALERKIN METHOD}

\textit{Analytical Preprocessing.} Whenever an analytical evaluation of the scalar products (A linear)
\[
( A u, \psi_k(\rho, \alpha))_{\rho, \alpha}
\]
is possible, it is done using the properties of the modified discrete Laguerre polynomials from Section 2.2. Examples for such a preprocessing can also be found in [18], [9], [52] and some results are used herein. The Galerkin equations (3.9) are assembled up to a truncation index \( n \). By means of a Gauss elimination the approximation \( u^n \) can be computed together with an error estimation \( \varepsilon_n \) due to Lemma 3.3. If the error is still too large, the Galerkin equations are extended by \( n \rightarrow n + 1 \).

For general problems, the coefficients \( a_k \) also depend on \( n \). Then the Galerkin equations are assembled up to an index \( \tilde{n} > n \) and the truncation error is controlled by (3.20).

\textit{Numerical Preprocessing.} The numerical evaluation of the scalar products (4.26) by the summation algorithm SUMMATOR (Chapter 6) requires some additional considerations. Following the red threat of this work, the tolerances for the summations have to be given adaptively, and the initial grids have to be chosen automatically.
Adaptive choice of the summation tolerance. The numerical evaluation of the scalar products (4.26) or (3.11) by the summation algorithm SUMMATOR (Chapter 6) requires a careful matching between the error control of the Galerkin process and the tolerance of the summation. Without information about the condition number of the linear system arising from the Galerkin equation, a sufficient summation tolerance cannot be given a priori. Even if for fixed truncation index \( n \) such a tolerance is known, an increase of \( n \) may increase the condition number, such that a summation performed for the old \( n \) is no longer sufficiently accurate. Then all summations have to be repeated. Thus the following algorithm is suggested:

Each new entry of the Galerkin equations is computed with a basic tolerance \( \tau_0^l \), on its own summation grid, which is stored. For fixed \( n \) the approximation \( \tilde{u}^n \) is computed for \( \tau_0^l \) and \( \tau_1^l := c \cdot \tau_0^l \), where \( 0 < c < 1 \) is a reduction factor. In actual computations \( c = 0.5 \) has been set. Comparison between the two results yields an estimation of the error \( \| u^n - \tilde{u}^n \| \), which can be regarded as a truncation error. The summation tolerance is increased until

\[
\| u^n - \tilde{u}^n \| < \frac{1}{2} \| \tilde{u}^n - \tilde{u}^{n-1} \|
\]

or \( c^k \tau_0^l < \tau_0^l \), \( \tau_0^l \) chosen before. If for \( n > n \) a refinement of a certain grid is required, this can be done by using the stored information.

Initial grids The choice of the initial grids for the SUMMATOR uses the only general insight known at the beginning of the summation: all terms in the sum of an \( H_{p,a} \) scalar product must be dominated asymptotically by the factor

\[
\rho^{s-1} = e^{-\lambda(s-1)}, \lambda > 0,
\]

from the weight function \( \Psi_{p,a} \). Thus the initial nodes are taken equidistant – in the logarithmic scale, where the actual value of \( \rho \) can give hints about the scale of the required upper bound of the sum. As pointed out in Chapter 6, the initial number of points has to be odd, the actual implementation uses 3–11 points with a controlling device.

Evaluation of a Galerkin Approximation. The pointwise evaluation of a Galerkin approximation

\[
u^n(s) = \sum_{k=0}^{n} a_k \psi_k(s; \rho, \alpha)
\]

has been realized by a fast and stable summation algorithm (adjoint summation) suggested by DEUFLHARD [14] for special functions satisfying a linear
homogeneous difference equation. As the basis functions $\psi_k$ can be generated by the three-term recurrence formula (2.22) started with

$$\psi_{-1} = 0, \quad \psi_0(s) = \Psi_{s,0}(s),$$

for given $s$ and $\Psi_{s,0}(s)$ the algorithm can easily be implemented. The computation of $\Psi_{s,0}(s)$, however, requires $s$ multiplications. For $\alpha \ll s$ computing time can be saved by recalling that (see e.g. [20], p. 190)

$$\lim_{s \to \infty} \frac{\Gamma(s + \alpha)}{s^{\alpha} \Gamma(s)} = 1, \quad \alpha > -1.$$ 

This means that for $\alpha \ll s$ we can substitute

$$\left(\frac{s - 1 + \alpha}{s - 1}\right) \rightarrow \frac{s^\alpha}{\Gamma(1 + \alpha)}.$$

An error estimation can be found in [52] (replacing $\alpha \rightarrow -\alpha$ there). This strategy has been used in Example 5.3 above, where $s$ can range up to $10^6$.

4.3 The Program CODEX

The implementation of the algorithm developed in this work is called CODEX (Countable systems of Ordinary Differential Equations by eXtrapolation). As the ideas of adaptivity used in the algorithm have a lot of structural similarities to packages developed at the Konrad-Zuse-Zentrum for the solution of partial differential equations, slightly modified parts of these programs could be implemented within CODEX. In particular the data structures play an important role for the development of adaptive algorithms. CODEX is written in the language C and consists of about 4000 lines of source code (80 Kbyte). The program is divided into four main parts:

(i) The time stepping module which is based on BORNEMANN’S KASTIX [8], extended by routines controlling the transformation in $H_{s,0}$-spaces (files: euler.c, extrapolation.c, transform.c).

(ii) The solution module, which performs Euler steps by means of the discrete Galerkin method, including assembling of the Galerkin equations, error estimation and control, handling of a family of basis expansions of solutions, numerical preprocessing and summation control, dynamic administration of the associated data structures. The solution of the arising (small) linear systems for the expansion coefficients is done by Gauss elimination (files: assemble.c, distbasic.c, distinter.c, estimate.c, gauss.c, solve.c).
(iii) The SUMMATOR, using data structures from the elliptic solver KASKADE by ROITZSCH [47], [48] and from KASTIX, adapted to the requirements of summation (files: summator.c, sumgrid.c).

(iv) Service routines for input, output, graphics including the pointwise evaluation of a solution $u(s)$.

The following rough flowchart documents the structure of the program CODEX. The numbers denote sections of this work, in which the modules of the program essentially have been developed or explained.

---

**Figure 4.1:** Flowchart of CODEX

---

63
5 NUMERICAL EXAMPLES

In this chapter some numerical results are presented featuring the following items:

- Adaptivity of the parameters $\rho$, $\alpha$
- Adaptivity of the the truncation index $n$
- Treatment of general problems and control of the truncation error
- Numerical preprocessing using the adaptive multilevel algorithm SUMMATOR
- Treatment of non-linear problems by means of the semi-implicit Euler discretization

All examples are excerpts from real life problems, in particular from polymer chemistry. All computations have been performed on a SPARC-station 1+ using double precision. The computing times (CPU) are given in seconds.

**Example 5.1: Backward difference equation.** The equation

$$ u'(t) = -\nabla u(t), \quad u(0) = \varphi $$

appears as a basic step in many applications in chemistry and is called chain addition process or propagation process there. It can also be considered as a stochastic process (e.g. random walk). As the operator $\nabla$ is bounded in $H_{\rho,\alpha}$ (compare Example 1.6 (i)), the problem is solved by use of the explicit Euler scheme.

The projections $P_{n}^{\rho,\alpha}$ can be performed in every time step (with variable truncation index $n$) using (3.14).

The process is started with the geometric distribution $\varphi = \Psi_{\infty,0}$, $\rho_0$ given. The choice $\rho_0 = 0.3$ is made here to illustrate the parameter control of the algorithm. From (1.5) the solution $u(t)$ is expected to be similar to a Poisson distribution with parameter $\lambda = t$. As the weight function $\Psi_{\rho,\alpha}$ tends to such a Poisson distribution for $\rho \to 0$ and $\alpha = t/\rho$ (Section 3.2), we can expect to obtain Galerkin approximations in spaces $H_{\rho,\alpha}$ with $\rho \ll \rho_0$ and $\alpha$ large. The algorithm has to transfer the initial geometric distribution to the (moving) Poisson distribution.

The system has been integrated up to $T = 50$ sec. It turns out, that the parameter $\rho$ decreases to $\rho(T) = 3.6 \cdot 10^{-3}$, whereas the parameter $\alpha$ increases from zero to $\alpha(T) = 13844.4$ – this is presented in Figure 1.1 (logarithmic scale). Note that $\rho(T)\alpha(T) \approx T$ as predicted in Section 3.2.
Table 5.1 reflects the nice behavior of the time-step control – the number of time steps remains nearly constant for increasing tolerance, whereas the order is increased.

<table>
<thead>
<tr>
<th>TOL</th>
<th>time-steps</th>
<th>max. order</th>
<th>( n_{\text{max}} )</th>
<th>norm of true error in ( H_{p,\alpha} )</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^{-1} )</td>
<td>30</td>
<td>3</td>
<td>4</td>
<td>( 3 \cdot 10^{-1} )</td>
<td>0.6</td>
</tr>
<tr>
<td>( 10^{-2} )</td>
<td>82</td>
<td>2</td>
<td>4</td>
<td>( 4 \cdot 10^{-2} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( *10^{-3} )</td>
<td>71</td>
<td>4</td>
<td>7</td>
<td>( 2 \cdot 10^{-5} )</td>
<td>1.8</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>69</td>
<td>5</td>
<td>11</td>
<td>( 8 \cdot 10^{-5} )</td>
<td>4.7</td>
</tr>
</tbody>
</table>

* run represented in Fig. 5.1-5.4

**TABLE 5.1:** CODEX: performance for variable order (Example 5.1)

The true error measured in \( H_{p,\alpha} = H_{p(T),\alpha(T)} \) has been obtained here from (1.6). Throughout this chapter \( n_{\text{max}} \) may denote the maximum of the number of expansion coefficients in \( H_{p(t),\alpha(t)} \) required to represent the solution after a global time step. The truncation index in single Euler steps sometimes is larger. Figure 5.2 shows the time layers chosen by the algorithm. The initial time-step size has been \( 10^{-3} \) and increases in time to a scale of 1. Recalling (1.39) we note that

\[
\| \nabla \|_{p,\alpha} \to \infty, \text{ } \alpha \text{ fixed, } \rho \to 0.
\]

This would result in time steps tending to zero in general cases. But insertion of \( \rho \alpha = t \) (the parameter of the Poisson distribution) shows, that in this case the step sizes may tend to one – an effect which actually has been turned out in the computations. In Figure 5.3 it can be observed, that with exception of the initial phase the time steps increase nearly linear.
The behavior of the error estimation is presented in Figure 5.4. At the beginning the transfer from geometric to a rough Poisson distribution leads to a little trouble, then a slight global error increase can be seen (the problem is non-stiff), but after a while the solution approaches to a Poisson distribution – the global error decreases.
Example 5.2: Free Radical Polymerization. We return to Example 1.4 now, where we discussed the behavior of the radical polymers during the initial phase of a free radical polymerization process. A (slightly simplified) modeling of the kinetics of the radical polymers leads to

\[
\begin{align*}
\dot{u}_1(t) &= -k_p(t) u_1(t) + f(t), \\
\dot{u}_s(t) &= -k_p(t)(u_s(t) - u_{s-1}(t)) - k_t u_s(t)
\end{align*}
\]

with \(u_s(t)\) denoting the concentration of the radical polymer \(s\). The reaction coefficients are

\[(5.1) \quad k_p(t) = e^{-0.01t}, \quad k_t = 5 \cdot 10^{-2}.
\]

An initiation process is modeled by

\[(5.2) \quad f(t) = I_0 k_d e^{-k_d t}, \quad I_0 = 1.0 \cdot 10^{-2}, \quad k_d = 10^{-1}.
\]

The reaction constants are chosen such that (5.1) reflects the most important structures of the complete polymerization system at the beginning of the reaction. As the solution tends to a nearly stationary state, the implicit Euler discretization has been used here. The analytical preprocessing is not changed by the time dependent reaction coefficients.

Figure 5.5 presents a simulation up to \(t = 100\) sec. This time is comparable to \(t = 5\) sec in the discussion of the original process (Example 1.4). The solution starts as a geometric distribution \((\rho = 0.1)\), approaches a Poisson distribution with parameter \(\lambda \sim t\), and finally returns to a geometric distribution with \(\rho \to 1\). This is one possible (but not the only) behavior of a free radical
polymerization discussed in Example 1.4. In between, around $t = 12$ sec., the solution has two local extrema and at most 14 expansion coefficients are required for performing a global time step (Figure 5.6). For an accuracy of TOL = 0.05 34 time steps are necessary, with increasing size at the end.

**Figure 5.5:** Time evolution of radical polymer (Example 5.2).

**Figure 5.6:** Time layer at $t = 12$ sec (Example 5.2).
True solution (---), Galerkin approximation (—).
Figure 5.7 presents the product \( \rho \alpha \), nicely reflecting the above interpretation of the process. Between \( t=0.5 \) sec and \( t=4 \) sec the product \( \rho \alpha \) increases nearly linear, because the solution is approximately a Poisson distribution in this phase and therefore \( \rho(t)\alpha(t) \approx t \).

\[ \int_0^t \alpha(t) \approx P(1,t) \]

\[ \rho(t) \approx P(\lambda,t) \]

\[ \alpha(t) \approx 1 - P(1,t) \]

\[ \rho(t) \approx \lambda \]

\[ \alpha(t) \approx P(\lambda,t) \]

\[ \int_0^t \alpha(t) \approx P(1,t) \]

\[ \rho(t) \approx P(\lambda,t) \]

\[ \alpha(t) \approx 1 - P(1,t) \]

\[ \rho(t) \approx \lambda \]

\[ \alpha(t) \approx P(\lambda,t) \]

\[ \int_0^t \alpha(t) \approx P(1,t) \]

\[ \rho(t) \approx P(\lambda,t) \]

\[ \alpha(t) \approx 1 - P(1,t) \]

\[ \rho(t) \approx \lambda \]

\[ \alpha(t) \approx P(\lambda,t) \]

\[ \int_0^t \alpha(t) \approx P(1,t) \]

\[ \rho(t) \approx P(\lambda,t) \]

\[ \alpha(t) \approx 1 - P(1,t) \]

\[ \rho(t) \approx \lambda \]

\[ \alpha(t) \approx P(\lambda,t) \]

\[ \int_0^t \alpha(t) \approx P(1,t) \]

\[ \rho(t) \approx P(\lambda,t) \]

\[ \alpha(t) \approx 1 - P(1,t) \]

\[ \rho(t) \approx \lambda \]

\[ \alpha(t) \approx P(\lambda,t) \]

\[ \int_0^t \alpha(t) \approx P(1,t) \]

\[ \rho(t) \approx P(\lambda,t) \]

\[ \alpha(t) \approx 1 - P(1,t) \]

\[ \rho(t) \approx \lambda \]

\[ \alpha(t) \approx P(\lambda,t) \]

\[ \int_0^t \alpha(t) \approx P(1,t) \]

\[ \rho(t) \approx P(\lambda,t) \]
Table 5.2 presents the performance of CODEX for variable tolerances in Example 5.2.

<table>
<thead>
<tr>
<th>TOL</th>
<th>time-steps</th>
<th>max. order</th>
<th>n_{max}</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-1}</td>
<td>21</td>
<td>2</td>
<td>7</td>
<td>0.7</td>
</tr>
<tr>
<td><em>5</em>10^{-2}</td>
<td>34</td>
<td>2</td>
<td>14</td>
<td>1.7</td>
</tr>
<tr>
<td>10^{-2}</td>
<td>55</td>
<td>2</td>
<td>29</td>
<td>14.5</td>
</tr>
</tbody>
</table>

* run represented in Fig. 5.5-5.8

TABLE 5.2: CODEX: performance for variable order (Example 5.2)

The increase of the computing time for accuracy $10^{-2}$ gives an hint, that the algorithm runs in difficulties, if the solution of a process is too far from the family $\Psi_{p,\alpha}$, e.g. distributions with two or more extrema or piecewise nearly constant solutions.

**Example 5.3: Polymer Degradation.** In a degradation reaction of the type

$$ (5.4) \quad P_s \xrightarrow{k_{sr}} P_r + P_{s-r} , \quad s > r \geq 1 , $$

a polymer $P_s$ of chain length $s$ breaks at position $r$ into two polymers of length $r$ and length $s-r$. In general (see e.g. [4]) the reaction rate coefficients $k_{sr}$ depend on the degree of the polymer $s$ and the location $r$ of the breaking bond in the polymer chain. Mathematical modeling of a degradation leads to the following CODE, with $u_s(t)$ the number of polymers of chain length $s$ at time $t$:

$$ (5.5) \quad u'_s(t) = (A_D u)(s) := - \left( \sum_{r=1}^{s-1} k_{sr} \right) u_s(t) + 2 \sum_{r=s+1}^{\infty} k_{sr} u_r(t) , $$

re-defining the degradation operator (1.42). A realistic initial distribution $u_s(0)$ is given in [52] by

$$ (5.6) \quad u_s(0) = \frac{s}{r} \tilde{p}^s , \quad \tilde{p} = e^{1/r} , $$

such that the maximum of the distribution $u_s(0)$ roughly occurs at chain length $s = r$. We consider the case $k_{sr} = k_p$ , $k_p$ constant first. For an analytical preprocessing we need the relation (2.33) and employ the recurrence formula (2.22). Then the scalar products

$$ a_{jk} := (A_D \psi_k , \psi_j)_{\rho,\alpha} $$
can be evaluated:

\[
(5.7) \quad a_{jk} = \begin{cases} 
\frac{k_p}{1 - \rho} j, & \text{for } k = j - 1, k \neq 0, \\
\frac{k_p}{1 - \rho} (-j - \rho(j - 1) - \alpha \rho), & \text{for } k = j, k \neq 0, \\
\frac{k_p}{1 - \rho} \rho(j - 1)(1 + \alpha/(j + 1)), & \text{for } k = j + 1.
\end{cases}
\]

For \(|j - k| > 1, k \neq 0\) the scalar products are zero (\(\rightarrow A_D\) is 1-invariant), the remaining terms are

\[
a_{00} = \frac{k_p}{1 - \rho} \rho(1 + \alpha), \quad a_{10} = \frac{k_p}{1 - \rho}(1 + \rho \alpha), \quad a_{j0} = \frac{2k_p \rho \alpha}{(1 - \rho)(j + 1)}, \quad j \geq 2.
\]

We know from Example 1.6 and Corollary 1.12, that the algorithm can be applied to the degradation operator for \(\rho(1 + \alpha/2) < 1\). Therefore we cannot use the exact representation (3.4) of the initial value \(\varphi\) by

\[
\varphi(s) = \frac{\bar{\rho}}{(1 - \bar{\rho})^2} \Psi_{\beta,1}(s), \quad s \geq 1,
\]

when \(\bar{\rho} > 2/3\) (\(\bar{\rho} > 1/2\) with a safety factor). This means, that distributions of the type (5.6) can only be represented in \(H_{\beta,1}\) for \(r = 1\). But with this we are back to a geometric distribution. Hence we use the representation of \(\varphi\) in \(H_{\beta,0}\) with \(\rho = 2\bar{\rho}/(1 + \bar{\rho})\) and the expansion coefficients \(a_k\) from Example 3.1 and have to control the moving weight function fitting additionally by \(\bar{\rho}(1 + \alpha/2) < c, c < 1\) a safety factor. Taking e.g. 30 coefficients introduces an error of about \(10^{-8}\) for \(\varphi\) – sufficient for all computations. By the way we note, that a simulation started in \(H_{\beta,1}\) failed indeed.

Figure 5.9 shows a typical simulation of a degradation up to \(t = 1000\) seconds, started with (5.6) having a maximum at \(r = 60000\) and a reaction constant \(k_p = 2.11 \cdot 10^{-7}\) due to [4]. The lines show the weight distribution

\[
P_s(t) := s \cdot u_s(t), \quad 1 \leq s \leq 400000.
\]

As the qualitative behavior of the process is the same for small \(r\), Figure 5.10 and Table 5.3 are computed using \(r = 50, k_p = 1\) and \(t = 1\). This numbers are chosen just to compare the results with a true solution obtained by direct integration of a system of 1000 ODE's here.
The error estimation works well again, what can be seen from Figure 5.10.

The performance of the algorithm for several tolerances is illustrated in Table 5.3.
As the degradation operator is not invariant in the sense of Definition 3.1, we also have to control the truncation error by increasing the actual truncation index $n$ to an $\tilde{n} > n$. Tests show, that in all steps $n + 1$ or $n + 2$ coefficients are sufficient to ensure the condition (3.20). In all runs the number of expansion coefficients decreases during the time evolution. As expected from the theory, the degradation operator is smoothing ($A_1$ is dissipative).

**Numerical Preprocessing.** Now the degradation problem will be attacked for

\[(5.8) \quad k_{sr} = k_s = k_p s^\beta, \quad k_p = 2.11 \cdot 10^{-7}, \quad \beta = -1/3.\]

This modeling of a heterogeneous polymerization is given in [4] and has been treated in [52] by replacing the fractional power by a so-called factorial power. Introducing a (small and controlled) modeling error, the problem could be solved there using product linerization formulas of discrete Laguerre polynomials. In order to solve the original problem, we use the SUMMATOR in order to evaluate the scalar products

\[(5.9) \quad \left(A_D^\alpha \psi_k(\rho, \alpha), \psi_j(\rho, \alpha)\right)_{\rho, \alpha} = k_p \sum_{s=2}^{\infty} \Psi_{\rho, \alpha}(s) \left(\frac{l_k(s)}{s} \right) \left(\frac{l_p(s)}{s+1} \right)\]

with $A_D^\alpha$ the degradation operator with coefficients (5.8) and the grid function $g(s)$ defined by

\[(5.10) \quad g(s) = -(s-1)l_j(s) + \frac{2}{1 - \rho} \left(\rho l_j(s) - l_j(s+1) + \rho^{j+1} \left(\frac{j + \alpha}{j + 1}\right)\right)\]

for $s > 2$ and

\[(5.11) \quad g(2) = -l_j(2) + 2l_k(2)l_j(1)\Psi_{\rho, \alpha}(2).\]

Here relation (2.33) enters again, but the analytic preparations cannot be continued, because there is no closed representation of the scalar products including the fractional power of $s$. Thus a numerical summation is necessary.

<table>
<thead>
<tr>
<th>TOL</th>
<th>time-steps</th>
<th>max. order</th>
<th>$n_{\text{max}}$</th>
<th>norm of true-error in $H_{\rho, \alpha}$</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>14</td>
<td>2</td>
<td>6</td>
<td>$3 \cdot 10^{-2}$</td>
<td>0.7</td>
</tr>
<tr>
<td>$5 \cdot 10^{-2}$</td>
<td>17</td>
<td>2</td>
<td>7</td>
<td>$2 \cdot 10^{-2}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$* \cdot 10^{-2}$</td>
<td>32</td>
<td>2</td>
<td>10</td>
<td>$7 \cdot 10^{-3}$</td>
<td>5.0</td>
</tr>
<tr>
<td>$5 \cdot 10^{-3}$</td>
<td>21</td>
<td>3</td>
<td>11</td>
<td>$3 \cdot 10^{-3}$</td>
<td>5.5</td>
</tr>
</tbody>
</table>

* run represented in Fig. 5.9, 5.10

**Table 5.3:** CODEX: performance for variable order (Example 5.3)
The simulation is started with \((5.6), r = 60000\), the upper bound \(s_{\text{max}}\) of the sum in the scalar product is chosen to be \(10^6\). This number is not crucial. Obviously the computational effort is now significantly higher than the effort in the case of constant reaction coefficients above, in particular for high accuracies and high time discretization order. Thus we restrict ourselves to technical accuracies and order 2 in the extrapolation table. Figure 5.11 compares the solution \(u(t)\) at \(t = 3600\) sec obtained by the method suggested in [52] (•••) and by CODEX (—). It can be seen that the results are in good agreement.

The number of nodes in the finest summation grid per time step is presented in Figure 5.12. The adaptive generation of these grids, described in Chapter 6, is controlled by the required truncation error of the Galerkin approximation, which in turn gets its tolerance from the time-step control. This means that the outline of the algorithm, suggested at the end of Section 1.1, has been completely realized.

![Figure 5.11: Heterogeneous degradation, t = 3600 sec. (Example 5.4)](image)

74
FIGURE 5.12: Evolution of the maximum number of grid points per step.

The following Table 5.4 demonstrates, that only about 45000 evaluations of the grid function \( g(s) \) are necessary during the whole computation (TOL=5 \( \times \) 10\(^{-2} \)).

<table>
<thead>
<tr>
<th>TOL</th>
<th>time-steps</th>
<th>( n_{\text{max}} )</th>
<th>max. no. of grid points</th>
<th>total no. of grid points</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^{-1} )</td>
<td>7</td>
<td>8</td>
<td>63</td>
<td>24258</td>
<td>22.3</td>
</tr>
<tr>
<td>( 5 \times 10^{-2} )</td>
<td>10</td>
<td>10</td>
<td>67</td>
<td>45125</td>
<td>41.1</td>
</tr>
</tbody>
</table>

* run represented in Fig. 5.11, 5.12

TABLE 5.4: CODEX: performance for numerical preprocessing (Example 5.3).

Example 5.4: Coagulation Processes. Finally an important and challenging nonlinear class of problems is addressed. Coagulation (combination) processes can be described in chemical notation by

\[ P_r + P_s \xrightarrow{k_{rs}} P_{r+s}, \]

where \( P_s \) may denote a polymer molecule or a soot (smog) particle of size \( s \). Coagulation (combination) processes appear frequently in applications - distinguished by different modelings of the reaction rate coefficients \( k_{rs} \). In polymer chemistry often moment dependent rate coefficients are in use, whereas the modeling of surface effects for the combination of smog particles leads to coefficients dependent on the size of the reacting molecules. The CODE of a coagulation process reads in general \( u_s(t) \) defined as in Example 5.3

\[ u'_s(t) = F(u(t)) := \frac{1}{2} \sum_{r=1}^{s-1} k_{rs-r} u_r(t) u_{s-r}(t) - u_s(t) \sum_{r=1}^{\infty} k_{sr} u_r(t). \]

75
The Frechét derivative $DF(\varphi)(u)$ of $F$ with respect to $u$ at $\varphi$ (required for a semi-implicit Euler discretization) can be computed pointwise (the time dependency is omitted) by

$$DF(\varphi)(u)(s) = \sum_{r=1}^{s-1} k_{r,s-r} \varphi_{s-r} u_r - \varphi_s \sum_{r=1}^{\infty} k_{s-r} u_r - u_s \sum_{r=1}^{\infty} k_{s-r} \varphi_r .$$

The treatment of a coagulation process with general rate coefficients $k_{sr}$ is a very difficult task, which has actually not been solved up to now. In [39] a discrete Fourier transform is applied, however, apart from the fact, that it turned out to be very time-consuming, a product separation of the $k_{rs}$ with respect to $r$ and $s$ has been assumed. This is not met in many examples – see e.g. the realistic coefficients

$$k_{r,s} := k_p \left( \frac{1}{r} + \frac{1}{s} \right)^{1/2} \left( r^{1/3} + s^{1/3} \right)^2 , \quad k_p \text{ constant},$$
given by Frenklach [21].

First, the behavior of a semi-implicit Euler discretization will be demonstrated for the case $k_{rs} = k_p$, $k_p$ constant. We assume $u, \varphi \in H_{p-\varepsilon,0}$ with expansions

$$u_s = \sum_{k=0}^{\infty} a_k \psi(s; \rho, 0), \quad \varphi_s = \sum_{k=0}^{\infty} b_k \psi_k(s; \rho, 0),$$
in $H_{p,0}$. For $\alpha = 0$ we can use the convolution formula (2.34). Then by manipulations essentially described in [18] the preprocessing results in

$$\frac{1}{\gamma^p} (DF(\varphi)(u), \psi_j)_p = \begin{cases} -k_p a_0 b_0, & j = 0, \\ k_p \sum_{k=1}^{j-1} a_k b_{j-k} - \frac{k_p}{\rho} \sum_{k=0}^{j-1} a_k b_{j-1-k}, & j \neq 0. \end{cases}$$

We end up with the following Galerkin equations for the semi-implicit Euler discretization ($a = (a_0, a_1, \ldots), b = (b_0, b_1, \ldots)$)

$$\begin{align*}
(I - \tau k_p C) a_\Delta &= \tau k_p f \\
a &= b + a_\Delta,
\end{align*}$$

where the Matrix $C = (c_{jk})$ is defined by

$$\begin{align*}
c_{00} &= -b_0, & c_{jk} &= 0, & j &\leq k, \\
c_{j0} &= -b_{j-1}/\rho, & c_{jk} &= b_{j-k} - b_{j-k-1}/\rho, & j &\geq 1, & c_{jk} &= b_{j-k} \quad \text{otherwise.}
\end{align*}$$
The right-hand side vector \( f \) is given by ([18], (4.14))

\[
\begin{align*}
    f_0 &= -\frac{1}{2} b_0^2, \\
    f_j &= \frac{1}{2} \left( \sum_{k=1}^{j-1} b_k b_{j-k} - \frac{1}{\rho} \sum_{k=0}^{j-1} b_k b_{j-1-k} \right), \quad j > 0.
\end{align*}
\]

An analytical preprocessing for \( \alpha \neq 0 \) is not possible here, because a relation like (2.34) is missing for that case. Fortunately this is no disadvantage here, as a check after each time step during the integration has revealed. During the whole process \( \bar{\alpha}(u) \) evaluated due to the fitting condition turned out to be near zero. Thus the integration has been performed with fixed \( \alpha = 0 \). Figure 5.13 shows a time layer of the weight chain length distribution for \( t = 1000 \) sec. process time, \( s_{\text{max}} = 4000, k_p = 1. \) The integration has been started with \( \varphi = \Psi_{p,0}, \rho = 0.001, \) i.e. \( \varphi \approx \delta_{s,1}. \)

**Figure 5.13**: Time layer of \( P_s(t) = s \cdot u_s(t) \) of a coagulation process at \( t = 1000. \)

Table 5.5 illustrates, how well the order and step-size control behaves even in the nonlinear case, when the semi-implicit Euler discretization is used.
Now we turn to a coagulation process from soot formation with the reaction rate coefficients (5.14). In [21], where the problem is approached by a method of moments, approximations of fractional-order moments based on the special form of these coefficients have been exploited to obtain information about the mean values of the solution at least. Nevertheless, the solution itself has to be computed by a direct time integration of a truncated system as an ODE ($\infty \to s_{\text{max}}$ in (5.12), $s_{\text{max}}$ large enough). This has been done in [32] for a comparable system with $s_{\text{max}} = 50, \ldots, 500$, leading to computing times up to 16000 sec (CPU) on a Cyber 205, increasing quadratically with $s_{\text{max}}$.

We present first results for such systems obtained by the discrete Galerkin method. In Table 5.6 and Figure 5.14 results of the nonstiff ODE - solver DIFEX1 [16] and CODEX (global tolerance $\varepsilon = 0.1$, $\alpha = 0$ fixed, $n_{\text{max}} = 3$) are compared for various process times. Obviously, the direct integration is very fast for small dimensions of the ODE system, but the computing time is going up quadratically after a while ($s_{\text{max}}$ enters quadratically in the evaluation of the right-hand side of (5.12)). For stiff integration, which is absolutely necessary for long time simulations, things become even worse. In contrast to that, the time spent by CODEX increases only linearly with $s_{\text{max}}$. As an explanation, Figure 5.15 shows the average number of nodes computed by the SUMMATOR in selected time steps and the number $s_{\text{max}}$ which is necessary to ensure reasonable solutions of the truncated system (5.12) at time $t$. The number of nodes in CODEX remains nearly constant during the whole integration, whereas a direct integration leads to increasing size of the systems. In the light of these facts, discrete Galerkin methods as suggested in this work seem to be the methods of choice for these problems. They do not depend on $s_{\text{max}}$ really, but only on the complexity of the solution to be represented by orthogonal polynomials, and as far as numerical preprocessing is applied, on the grids generated by the summation algorithm. Even if a careful and extensive examination of heterogeneous coagulation processes is left to future work, the first results are very promising in contrast to other methods used up to now.

<table>
<thead>
<tr>
<th>TOL</th>
<th>time-steps</th>
<th>max. order</th>
<th>$n_{\text{max}}$</th>
<th>true error in $H_{p,q}$</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>32</td>
<td>2</td>
<td>3</td>
<td>$1.1 \cdot 10^{-1}$</td>
<td>0.8</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>71</td>
<td>4</td>
<td>3</td>
<td>$1.5 \cdot 10^{-2}$</td>
<td>1.9</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>63</td>
<td>4</td>
<td>3</td>
<td>$1.1 \cdot 10^{-3}$</td>
<td>2.3</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>64</td>
<td>5</td>
<td>3</td>
<td>$8.6 \cdot 10^{-5}$</td>
<td>4.0</td>
</tr>
</tbody>
</table>

* run represented in Fig. 5.13

Table 5.5: CODEX: performance for several tolerances (Example 5.4).
TABLE 5.6: Comparison between direct (non-stiff) integration (DIFEX1) and CODEX.

<table>
<thead>
<tr>
<th>$t$</th>
<th>$s_{\text{max}}$</th>
<th>CPU—DIFEX1 (sec)</th>
<th>CPU—CODEX (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>20</td>
<td>3</td>
<td>51</td>
</tr>
<tr>
<td>0.5</td>
<td>50</td>
<td>68</td>
<td>170</td>
</tr>
<tr>
<td>1.0</td>
<td>85</td>
<td>329</td>
<td>280</td>
</tr>
<tr>
<td>2.0</td>
<td>180</td>
<td>1158</td>
<td>525</td>
</tr>
</tbody>
</table>

FIGURE 5.14: Comparison between direct solution (—) and Galerkin approximation (—*) of a heterogeneous coagulation process at $t=2$. 
Figure 5.15: Time evolution of $s_{\text{max}}$ and average number of nodes of summation grids.
Appendix: A Multilevel Summation Algorithm

As pointed out in the Chapters 3 and 5, the evaluation of scalar products in $H_{p,0}$ sometimes has to be done numerically. The task is the computation of sums of the form

$$\sum_{s=1}^{s_{\text{max}}} u_s l_j(s; \rho, \alpha)$$

arising from the Galerkin method based on modified discrete Laguerre polynomials, or of sums

$$\sum_{s=s_{\text{min}}}^{s_{\text{max}}} f(s)$$

in general for a given grid function $f(s)$, $s \in [s_{\text{min}}, s_{\text{max}}]$, $s \in \mathbb{N}$. In order to obtain an approximation of (6.2) a so-called SUMMATOR has been developed. The idea is the construction of a sequence of grids

$$\Delta^0 \subset \Delta^1 \subset \ldots \subset \Delta^j \subset [s_{\text{min}}, s_{\text{max}}],$$

which should reflect the behavior of the grid function $f$. Therefore we need local summation formulas, a local error estimation and a refinement strategy. The algorithm might have applications apart from the context of this work.

Discrete Interpolation. Interpolation of a sequence (grid function) $f(s)$ only makes sense, if we assume an underlying smoothness of $f$. The error of polynomial interpolation for functions $f \in C^k[a, b]$ can be described by means of the $k$-th derivative of $f$. In the discrete case we have to replace the derivative by a finite difference. Thus the interpolation error will contain all values of a grid function $f(s)$. For a given subset

$$I := \{s_{\text{min}}, s_{\text{min}} + 1, \ldots, s_{\text{max}}\} \subset \mathbb{N}$$

let

$$\Delta := \{s_{\text{min}} = s_0 < s_1 < \ldots < s_n = s_{\text{max}}\} \subset I$$

a sub-grid, where the discrete interpolation polynomial $D_n(s) = D_n^{s_0 \ldots s_n}(s)$ of degree $n$ is defined by the conditions

$$D_n^{s_0 \ldots s_n}(s_i) = f(s_i), \ s_i \in \Delta.$$ 

As in the continuous case $D_n(s)$ can be constructed by means of divided differences $f[s_i, \ldots, s_{i+k}]$ [51]. The polynomial $D_n(s)$ then reads

$$D_n^{s_0 \ldots s_n}(s) = \sum_{j=0}^{n} \omega_j(s)f[s_0, \ldots, s_j],$$

$$\omega_j(s) := (s - s_0) \ldots (s - s_{j-1}),$$

for $s \in \Delta$. 

81
Lemma 6.1. For \( r \in I \) the interpolation error \( e^r_j(\tau) \) is given by

\[ e^r_j(\tau) := f(\tau) - D_n(\tau) = \omega_{n+1}(\tau) f[s_0, \ldots, s_n, \tau]. \]

Proof. The representation of \( D_n^{s_0, \ldots, s_n}(s) \) in terms of divided differences leads to

\[ f(\tau) - D_n^{s_0, \ldots, s_n}(\tau) = D_{n+1}^{s_0, \ldots, s_n}(r) - D_n^{s_0, \ldots, s_n}(r) \]
using (6.4).

Lemma 6.1 shows, that the 'smoothness' of grid functions is measured in the present context by the value of the maximum of the divided differences \( f[s_0, \ldots, s_n, r] \) on \( I \). The theorem ensures that for a moderate bound of this difference the behavior of the interpolation will be comparable to the continuous case.

Summation formulas. The construction of summation rules can be done analogue to the derivation of quadrature formulas. For given nodes \( s_0, \ldots, s_n \), the interpolation polynomial \( D_n^{s_0, \ldots, s_n}(s) \) is summed from \( s = s_0 \) to \( s = s_n \).

Discrete Trapezoidal rule. Consider the discrete interval \( \{s_0, s_0+1, \ldots, s_1\} \)
with \( s_1 := s_0 + \delta, \delta \in \mathbb{N} \). Let \( f_0 := f(s_0) \) and \( f_1 := f(s_1) \). Then the trapezoidal sum \( T(\delta) \) of \( \sum_{s=s_0}^{s_1} f(s) \) turns out to be

\[ T(\delta) = \frac{\delta + 1}{2} (f_0 + f_1). \]

The difference to the continuous Trapezoidal rule arises from the fact, that edge points have to be considered separately. Insertion of the interpolation error from Lemma 6.1 lets expect the summation error \( \epsilon_\delta \) to behave like

\[ \epsilon_\delta \approx C \delta, \quad C \text{ constant}. \]

Note that for a repeated application of a summation rule the role of the bounds of the sub-intervals has to be kept in mind.
Discrete Simpson rule. Now take three nodes, \( s_0, s_1, s_2 \), where \( s_1 := s_0 + \delta_1 \) and \( s_2 := s_1 + \delta_2 \). As the local error estimation will compare Trapezoidal sums on an interval with Simpson sums we are interested in approximations \( S^L(\delta_1, \delta_2) \) and \( S^R(\delta_1, \delta_2) \) of \( \sum_{s=s_0}^{s_2} f(s) \) and \( \sum_{s=s_1}^{s_2} f(s) \). Some tedious calculations (which fortunately could have been performed with the algebraic package REDUCE) lead to the following expressions:

\[
S^L(\delta_1, \delta_2) = f_0 \frac{(2\delta_1 + 3\delta_2 + 1)(\delta_1 + 1)}{6(\delta_1 + \delta_2)} + f_1 \frac{(\delta_1 + 3\delta_2 - 1)(\delta_1 + 1)}{6\delta_2} - f_2 \frac{\delta_1(\delta_1^2 - 1)}{6\delta_2(\delta_1 + \delta_2)}
\]

(6.8)

for the left interval \( \{s_0, \ldots, s_0 + \delta_1\} \) and

\[
S^R(\delta_1, \delta_2) = -f_0 \frac{\delta_2(\delta_2^2 - 1)}{6\delta_1(\delta_1 + \delta_2)} + f_1 \frac{(3\delta_1 + \delta_2 - 1)(\delta_2 + 1)}{6\delta_1} + f_2 \frac{(3\delta_1 + 2\delta_2 + 1)(\delta_2 + 1)}{6(\delta_1 + \delta_2)}
\]

(6.9)

for the right interval \( \{s_1, \ldots, s_1 + \delta_2\} \).

The above formulas are the basis of the SUMMATOR.

Error estimation. Let \( I_j^i \) and \( I_{j+1}^i \) be sub-intervals of \( I \) which have been generated on refinement level \( i \) by subdividing an interval \( I_k^{i-1} \). We denote by \( T_j^i \) the Trapezoidal sum on \( I_j^i \) and by \( S_j^i \) the Simpson sum using the three different \( f \) - values the end points of \( I_j^i \) and \( I_{j+1}^i \) and the formula (6.8) (analogue for the right sub-interval).

Then the error

\[ |T_j^i - \sum_{I_j^i} f(s)| \]

will be estimated by

\[ \epsilon_j^i := |T_j^i - S_j^i|, \]

assuming that the Simpson formula will be significantly more accurate on \( I_j^i \) than the Trapezoidal rule.
Refinement strategy [5]. It is reasonable to equilibrate the local errors on each grid $\Delta^i$. Let $\Delta^{i-1}$ and $\Delta^i$ be constructed with $n_{i-1}$ respectively $n_i$ subintervals and errors $\varepsilon^{i-1}_{k}$ and $\varepsilon^i_j$. If we refine an interval $I^i_j$ again, we can expect from (6.7) that the errors $\varepsilon^{i+1}_m$ and $\varepsilon^{i+1}_{m+1}$ on the new generated intervals $I^{i+1}_m$ and $I^{i+1}_{m+1}$ will be approximately

$$\varepsilon^{i+1}_j := \frac{(\varepsilon^{i-1}_k)^2}{\varepsilon^i_j},$$

under the assumption, that the associated intervals have been bisected. Then we compute a cut value by

$$\text{cut}_{i+1} = \max_{j=1, \ldots, n_i} \varepsilon^{i+1}_j,$$

and intervals $I^i_j$ with $\varepsilon^i_j \geq \text{cut}_{i+1}$ will be refined only.

Algorithm. In short form the algorithm reads as follows:

1. Choice of a start grid with an odd number of points (at least three), i.e an even number of subintervals.
2. Approximation of the sums on all intervals with discrete Trapezoidal and Simpson rule. Computation of the global sum, refinement of all intervals.
3. Local and global error estimation, where the global error is estimated by comparison of the approximations of the global sums on the last two refinement levels.
4. if global error (scaled) $< TOL$ : stop.
   if not: refinement of all intervals with error larger than the cut value.
5. Evaluation of the new sums; back to (3).

If the value of the sum turns out to be small ($< 1$), an absolute scaling is used. The algorithm has been implemented in the language C, using concepts and data structures from the finite element codes KASKADE [47], [48] and KASTIX [7].

Example 6.1. As a first example we consider the computation of the truncated harmonic series

$$\sum_{s=1}^{s_{\text{max}}} \frac{1}{s}.$$

This sum is a challenging test for the algorithm, because the harmonic series diverges logarithmically in $s_{\text{max}}$. So we have to expect, that the nodes chosen by
the SUMMATOR are nearly equidistant in a logarithmic scale. This is backed indeed by Figure 6.1, where the nodes for a summation up to \( s_{\text{max}} = 10^7 \) are shown for required tolerances \( \text{tol} = 10^{-2} \) and \( \text{tol} = 10^{-4} \). When \( s_{\text{max}} \) is increased to higher values, the distribution of the nodes remains qualitatively the same. The reached true error and the number of chosen nodes are presented in Table 6.1 for different tolerances. The exact value of the sum with \( s_{\text{max}} = 10^7 \) has been computed on a SPARC station 1+ (90 sec CPU time). The computing time of the SUMMATOR has been maximal 1 sec (CPU) for the most time-consuming test (2140 nodes).

<table>
<thead>
<tr>
<th>tol</th>
<th>error</th>
<th>nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^{-2} )</td>
<td>( 1 \cdot 10^{-3} )</td>
<td>47</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>( 5 \cdot 10^{-5} )</td>
<td>129</td>
</tr>
<tr>
<td>( 10^{-6} )</td>
<td>( 6 \cdot 10^{-7} )</td>
<td>356</td>
</tr>
<tr>
<td>( 10^{-8} )</td>
<td>( 2 \cdot 10^{-10} )</td>
<td>2140</td>
</tr>
</tbody>
</table>

**Table 6.1:** Performance of the SUMMATOR for the harmonic series.

**Figure 6.1:** Grids for the summation of the harmonic series in a logarithmic scale. Coarse grid Tol = \( 10^{-2} \) (47 nodes), fine grid TOL = \( 10^{-4} \) (129 nodes).

**Example 6.2.** The solution of CODE's by means of the discrete Galerkin method requires the evaluation of scalar products like \( (Au, \psi_n)_{\rho,\alpha} \), with \( A \) an operator from a CODE and \( \psi_n \) a basis function of \( H_{\rho,\alpha} \). Whenever an analytic calculation is not possible, the scalar products have to be computed numerically.
The present example will demonstrate, that this task can be efficiently accomplished by the SUMMATOR. For demonstration purposes we restrict ourselves here to a case which also can be treated analytically. For this we consider

\[(\psi_{10}(s; \rho, 0), \psi_{10}(s; \rho, 0))_{\rho, 0} = \rho^{10} \text{ and } (\psi_{10}(s; \rho, 0), \psi_{0}(s; \rho, 0))_{\rho, 0} = 0.\]

In both cases the terms of the sum are dominated by a polynomial of high degree. The second case is of interest as a test of the absolute scaling for results near zero. For the computations the sums in the scalar products have been truncated at \(s_{\text{max}} = 10^5\) for \(\rho = 0.999\).

**FIGURE 6.2:** Typical grid (tol= \(10^{-3}\), 63 nodes) arising from the summation of scalar products in \(H_{p,a}\).

The number of nodes for these examples range from 50 for tolerance \(10^{-2}\) up to about 500 for tolerance \(10^{-6}\). For further applications see Example 5.3.

**Example 6.3.** Finally we present some results for double series. We consider the sum

\[
\sum_{s=1}^{s_{\text{max}}} \sum_{r=1}^{s-1} l_1(s; \rho, 0) k_{r,s-r} P_r P_{s-r},
\]

with the coefficients (5.14)

\[
k_{r,s} := \left(\frac{1}{r} + \frac{1}{s}\right)^{1/2} \left(r^{1/3} + s^{1/3}\right)^2
\]
and

\[ P_s := s \rho^{t-1} . \]

This is the problem setting from the simulation of smog reactions in Example 5.4. The parameters \( \rho = 0.995 \) and \( s_{\text{max}} = 1000 \) have been chosen here, such that a direct summation could be performed in moderate computing times. Table 6.2 shows the number \( n_f \) of evaluations of the term in the inner sum for technical accuracies. Note that this number is about 500000 if the sum is computed directly. Other successful attempts have been made for double sums arising in degradation processes.

\[
\begin{array}{|c|c|c|}
\hline
\text{tol} & \text{error} & n_f \\
\hline
10^{-1} & 1 \cdot 10^{-2} & 93 \\
10^{-2} & 6 \cdot 10^{-3} & 203 \\
10^{-3} & 2 \cdot 10^{-4} & 1072 \\
10^{-4} & 4 \cdot 10^{-5} & 1729 \\
\hline
\end{array}
\]

Table 6.2: Performance of the SUMMATOR for a double series.

The choice of the required tolerance \( TOL_i \) for the inner sum is a difficulty, which also arises for quadrature of functions (see [38]). We use the following argumentation:

Given a sub-interval on level \( i \) of the outer sum, we need the \( f \)-value of the midpoint (used for computation of the Trapezoidal rule) with an error small enough to ensure the required global accuracy \( TOL \), if level \( i + 1 \) is the final one, but not smaller than the predicted error \( \varepsilon_i^{i+1} \) of the associated interval on the next level. Finally it is reasonable to require at least the global tolerance. Thus we obtain a local inner tolerance by two steps:

(i) \[ TOL_i = \max \left\{ \frac{TOL}{n^j}, \varepsilon_j^{i+1} \right\} , \]

(ii) \[ TOL_i = \min \{ TOL, TOL \} . \]

We also want to mention, that the effect of loss of smoothness described in [38] for the approximation of double integrals by an adaptive algorithm is not so dramatic here, since the divided differences remain moderately bounded if the grid function \( f \) is perturbed.
REFERENCES


