A Symbolic Algorithm for Solving Two-Point BVPs on the Operator Level [⋆]

Markus Rosenkranz [a,*], Bruno Buchberger [a], Heinz W. Engl [b]

[a] *Johannes Kepler University, Research Institute for Symbolic Computation, A-4040 Linz, Austria*

[b] *Johannes Kepler University, Institute for Industrial Mathematics, A-4040 Linz, Austria*

Abstract

We present a new method for solving regular boundary value problems for linear ordinary differential equations. Unlike existing methods that reduce everything to the functional level via the Green's function, our approach works on the level of operators throughout. We proceed by representing the operators needed as noncommutative polynomials using as indeterminates basic operators like differentiation, integration and boundary evaluation.

The crucial step for solving the boundary value problem is to understand the desired Green's operator as a suitable oblique Moore-Penrose inverse. The resulting equations are then solved for the Green's operator using a carefully compiled noncommutative Gröbner basis that reflects the essential interactions between basic operators.

We have implemented our method as a Mathematica$^{\text{TM}}$ package, embedded in the TH∃OREM∀ system developed in the group of the second author. We show some computations performed by this package.

Key words: Boundary value problems, differential equations, operator calculus, noncommutative Gröbner bases.

[⋆] This work was supported by the Austrian Science Foundation *FWF* under the SFB grants F1302 and F1308.

[*] Corresponding author. Tel: ++43 732 2468 9926. Fax: ++43 732 2468 9930.
Email addresses:
Markus.Rosenkranz@risc.uni-linz.ac.at (Markus Rosenkranz),
Bruno.Buchberger@risc.uni-linz.ac.at (Bruno Buchberger),
engl@indmath.uni-linz.ac.at (Heinz W. Engl).

The present article essentially summarizes the main points of the first author's PhD thesis (Rosenkranz, 2003), whose primary and secondary advisors are the second and third author, respectively. Some early ideas were published in (Rosenkranz, Buchberger, Engl, 2003), albeit on a purely heuristic basis without implementation: noncommutative Gröbner bases were computed by the MMA package *NCAlgebra* from UCSD (Helton, Miller, 2003; Helton, Stankus, Wavrik, 1998) on a per-problem basis rather than using a fixed Gröbner basis as presented here. A sketchy overview of (Rosenkranz, 2003) was presented by the first author in a poster at ISSAC'03 that is scheduled to appear as a four-page survey in the SIGSAM bulletin; naturally, this publication does not contain any algorithms, theorems or proofs.

1 Introduction

1.1 The Notion of Boundary Value Problem

In the engineering world, *boundary value problems* are ubiquitous: A physical process is typically described by (ordinary or partial) differential equations, where suitable side conditions serve to select the desired solution from the manifold of all solutions. The side conditions usually specify data where it can be measured or controlled easily—if time is involved, this may be the starting configuration (initial value problem, IVP); in case of space-based problems, it is usually some boundary data (boundary value problem, BVP).

Let us make this precise for the case of linear ordinary differential equations (LODEs) in one dimension. We are given a finite real interval $[a, b]$ and a *linear differential operator* T defined by

$$T\,u = c_n\,u^{(n)} + \ldots + c_1\,u' + c_0\,u, \tag{1}$$

for all $u \in C^\infty[a, b]$. For the moment, we will restrict ourselves to a completely smooth setting, so the coefficient functions $c_0, \ldots, c_n$ are also assumed to be in $C^\infty[a, b]$; see the remarks in 2.5 for extending this to the C^n or distributional case. Analytically, the operator T is naturally described as acting on the Banach space $(C[a, b], ||\cdot||_\infty)$ with dense domain of definition $C^n[a, b]$; see for example Engl, Nashed (1981). For our purposes, however, it will be sufficient to stick to a purely algebraic setting, where the domain of T is $C^\infty[a, b]$, viewed as a "naked" vector space. Besides that, it is convenient to understand $C^n[a, b]$ and $C^\infty[a, b]$ in a complex setting: the functions have type $[a, b] \to \mathbb{C}$, and the scalars range over $\mathbb{C}$.

Given a forcing function $f \in C^\infty[a, b]$, we want to solve $T\,u = f$ for $u \in C^\infty[a, b]$ subject to appropriate *side conditions*: In the case of initial conditions, we stipulate $u(a) = u_0, u'(a) = u_1, \dots, u^{(n-1)}(a) = u_{n-1}$ for given $u_0, u_1, \dots, u_{n-1} \in \mathbb{C}$. Turning to boundary conditions, we introduce for each $i = 1, \dots, n$ a boundary operator

$$\begin{aligned} B_i\ u = p_{i,0}\, u^{(n)}(a) + \ldots + p_{i,n-1}\, u'(a) + p_{i,n}\, u(a) \\ + q_{i,0}\, u^{(n)}(b) + \ldots + q_{i,n-1}\, u'(b) + q_{i,n}\, u(b), \end{aligned} \tag{2}$$

where the coefficient functions $p_{i,j}, q_{i,j}$ are again from $C^\infty[a,b]$. The corresponding boundary conditions are then $B_1\, u = b_1, \dots, B_n\, u = b_n$ with given $b_1, \dots, b_n \in \mathbb{C}$. Obviously, initial conditions are a special case of boundary conditions: they occur when all $q_{i,j}$ vanish and all $p_{i,j}$ are set to unity.

Hence the traditional *distinction* between IVPs and BVPs may appear strange to a person coming from the world of symbolic computation. There are, however, two good reasons for keeping them separate from each other:

- Their *qualitative nature* is vastly different: We know that IVPs are guaranteed to be solvable by Peano's Theorem as soon as the differential equation is just continuous. But BVPs may not always be solvable even for a linear differential equation; in fact, the issue of their solvability is intimately connected with the Banach space theory of eigenvalues.
- Whereas IVPs are viewed as "one-shot" computations, one usually considers BVPs as *parametrized* in the forcing function f.

Although these two issues are actually orthogonal to each other, in the literature, they are typically combined in way described.

In symbolic computation, we can often ignore the qualitative difference between IVPs and BVPs; we will also do so in this article. But the issue of parametrization is crucial for us: It means that we understand a BVP like

$$\begin{aligned} &T\ u = f, \\ &B_1\, u = b_1, \ldots B_n\, u = b_n. \end{aligned} \tag{3}$$

as having a *symbolic parameter* f. As soon as one instantiates f by some concrete function like sin, one ends up with an inhomogeneous differential equation (together with some side conditions for fixing the integration constants). Of course computer algebra has nowadays plenty of powerful algorithms attacking such problems—even if the underlying equation is substantially more complicated. However, virtually all such methods rely on concrete terms for denoting the inhomogeneity f, and they are often tuned to special classes of such functions.

Hence it is more appropriate to understand (3) as an *operator problem*: Given the differential operator T, boundary operators $B_1, \dots, B_n$ and complex numbers $b_1, \dots, b_n$, we want to find the operator G mapping an arbitrary $f \in$

$C^\infty[a,b]$ to the solution u of (3); in the literature (Stakgold, 1979), this is known as the Green's operator of the BVP. Note that G may have a perfectly simple algebraic representation even though u itself may have no representation at all. (This is already clear on grounds of cardinality: Symbolic solutions are bound to be countable, whereas $C^\infty[a,b]$ is obviously not so.)

1.2 What Is an "Operator-Based" Method?

The solution method we want to present here works on the *operator level* as announced in the title of this article. What do we mean by this? As explained in the previous section, the natural interpretation of a BVP like (3) is in terms of operators: Whereas solving differential equations means to search for objects of type $F \equiv \mathbb{R} \to \mathbb{C}$, a BVP amounts to fining an operator, i.e. as object having the higher type $F \to F$. Therefore we would prefer a calculus that yields the Green's operator G for (3) by performing calculations on various *operators* related to it.

As an alternative to such an operator-based method, one can translate the whole problem to a purely *functional setting*—and this is precisely what we find in the literature (Kamke, 1983, pages 188–190). The crucial idea for this is the following: In the case considered here, G can always be written as an integral operator having the so-called Green's function g as its kernel; see (Coddington, Levinson, 1955). So

$$Gf\,(x) = \int_a^b g(x,\xi)\,f(\xi)\,d\xi \tag{4}$$

for all $f \in C^\infty[a,b]$ and $x \in [a,b]$. Hence the problem of searching the *operator* G is reduced to finding the *function* g. (As we will see in the next section, our method also extracts the Green's function g in a kind of postprocessing step. However, this step is optional and may be seen as a concession to the traditionally function-based language of BVPs.)

While the classical translation approach does have its merits, we can see some unique *advantages* in our new approach:

- The operator-based approach has a greater *potential of generalization.* For example, the whole theory of Green's functions presupposes linear differential operators, and it is much less conspicuous for partial differential equations. (Of course, our method as stated here cannot be applied to these problems either. However, we see some chances of adaptation; see Section 4 for a brief discussion.)
- From a *conceptual viewpoint*, it is more satisfying to solve a problem at the level where it is actually stated. Of course, in mathematics we often

proceed by translating some problem into another, but uniform solution methods have the additional benefit of structural simplicity and clarity.

- Besides this, our method may be better in terms of *complexity.* We have not yet embarked on a rigorous analysis of this issue, but there are some indications pointing in this direction: The formula given in (Kamke, 1983, pages 189) involves Gaussian elimination with functional entries. At least for the important special case of differential operators with constant coefficients (see Subsection 2.4), our approach seems to be significantly faster.

1.3 Previous Work

Exact *solution methods* for linear BVPs are of course not new as we have pointed out (Kamke, 1983; Coddington, Levinson, 1955; Stakgold, 1979). But as far as we know, all these methods typically work on a functional level as explained in Subsection 1.2.

Operator-based methods are routinely used in symbolic summation and integration of *holonomic functions*; see (Zeilberger, 1990; Chyzak, Salvy, 1997). Noncommutative Gröbner bases are applied there for elimination in Ore algebras of operators. Unlike for solving BVPs, though, the issue there is searching for sequences/functions, which are described by suitable annihilation operators.

Originally we got the inspiration for our method from the paper (Helton, Stankus, Wavrik, 1998), which describes the use of noncommutative Gröbner bases for *simplifying* huge terms of operator control theory. Using a lexicographic term ordering, however, one can also employ noncommutative Gröbner bases for solving systems of operator equations, and this is essentially what we did on a per-problem basis in our early paper (Rosenkranz, Buchberger, Engl, 2003).

1.4 Structure of the Article

In Section 2, we describe our new solution method for BVPs in detail. Subsection 2.1 explains the

2 The Solution Method

2.1 General Setup

The solution method to be described applies to BVPs of the form (3), subject to the following restrictions:

- We assume that the BVP is *regular* in the sense that there must be unique solution. This implies that the boundary conditions be consistent and linearly independent.
- We will only cover the *semi-inhomogeneous* case, meaning that $b_1, \ldots, b_n$ are zero. This involves no loss of generality because any fully inhomogeneous problem can be decomposed into such a semi-homogeneous one and a rather trivial BVP with homogeneous differential equation and inhomogeneous boundary conditions; see (Stakgold, 1979, page 43).

First of all, let us set up an *operator-theoretic formulation* of (3). Using the Green's operator G, we have to fulfill $T\,Gf = f$ and $B_1\,Gf = \cdots = B_n\,Gf = 0$ for all $f \in C^\infty[a,b]$. By definition, this is equivalent to the corresponding operator equations $T\,G = 1$ and $B_1\,G = \cdots = B_n\,G = 0$.

Before we proceed, we establish the following *implicit lambda convention*. Whenever we use a term T (usually but not necessarily involving the variable x) in place of a function, we mean the mapping $x \mapsto T$ or, in computer-science notation, the lambda term $\lambda x.T$. The differentiation operator D acting on functions then corresponds to the operation usually denoted by ∂/∂_x.

In order to apply computer algebra methods, we interpret the involved operators as noncommutative polynomials.[1] For example, consider the *differential operator* in informal notation $T = x^3\,D^2 + e^x\,D + \sin x$. The coefficient functions $c_2 = x^3, c_1 = e^x, c_0 = \sin x$ can be seen as *multiplication operators* in the following sense: Any $f \in C^\infty[a,b]$ induces an operator M_f defined by $M_f\,u = fu$ for all $u \in C^\infty[a,b]$. Using this notation, the above operator can be written as $T = \lceil x^3 \rceil\,D^2 + \lceil e^x \rceil\,D + \lceil \sin x \rceil$, where juxtaposition denotes operator composition (note that this is consistent with the power notation for differentiation) and $\lceil f \rceil$ is a shorthand for M_f. In this way, any linear differential operator can be written as a noncommutative polynomial in the indeterminates D and M_f with f ranging over a certain functional domain yet

[1] In the sequel, we will not make any explicit distinction between a noncommutative polynomial and its "interpretation" as a linear operator on $C^\infty[a,b]$ or between an indeterminate and the corresponding "basic operator". The context will always make it clear what we mean. If the reader desires a more rigorous treatment, he may consult the PhD thesis of the first author (Rosenkranz, 2003).

to be fixed.

Turning to *boundary operators*, we have to introduce two more indeterminates. For the above operator T, a typical boundary operator could be $B_1 u = 2u'(a) - 3u(a) + 7u'(b)$. Let us write L and R for evaluation at the left and right boundary, respectively, so $L\,u = u(a)$ and $R\,u = u(b)$ for all $u \in C^\infty[a,b]$ (note that by the implicit lambda convention, these boundary operators actually map functions to functions, namely the constant functions having the corresponding boundary value). With this notation, the boundary operator under consideration would be represented by the noncommutative polynomial as $B_1 = 2L\,D - 3L + 7R\,D$

It is now clear how to formulate the differential and boundary operators of (3) as noncommutative polynomials in the indeterminates D, M_f, L, R. But this will clearly not be sufficient for representating the operator G that is supposed to solve (3), since the latter must involve *integration.* Hence we introduce the following operator A for computing the antiderivative

$$Af = \int_a^x f(\xi)\,d\xi$$

of any function $f \in C^\infty[a,b]$. Since we know that the n-th derivative of the Green's function in G jumps along the diagonal, we have to include the dual of A, namely the operator

$$Bf = \int_x^b f(\xi)\,d\xi,$$

such that the integral (4) can be patched accordingly by adding A and B portions (see Section 3 for examples).

We can now introduce the necessary *polynomial ring* formally. The domain $\mathfrak{F}$ used for parametrizing the multiplication operators will be introduced in Subsection 2.5. For the moment, it is sufficient to think of it as the $\mathbb{C}$-algebra $\mathfrak{Exp}$ with basis $\mathfrak{Exp}^\# = \{x^n e^{\lambda x} | n \in \mathbb{N} \wedge \lambda \in \mathbb{C}\}$; we call this the polyexponential algebra $\mathfrak{Exp}$. (Every algebra $\mathfrak{A}$ considered here is assumed to include the notion of a distinguished basis referred to as $\mathfrak{A}^\#$. An algebra that is at the same time a field will be called a field algebra.)

Definition 1 *Let $\mathfrak{F}$ be an analytic algebra. Then the noncommutative polynomial ring*

$$\mathbb{C}\langle\{D, A, B, L, R\} \cup \{M_f | f \in \mathfrak{F}^\#\}\rangle$$

will be called the ring of analytic polynomials *over $\mathfrak{F}$, denoted by $\mathfrak{An}(\mathfrak{F})$.*

Strictly speaking, we should from now on distinguish between the *formal operators* in $\mathfrak{An}(\mathfrak{F})$ and the *actual operators* in $\{U \,|\, U : C^\infty[a,b] \to C^\infty[a,b]\}$. Most of the time, however, it is either clear which of the two concepts we mean or a certain statement is true for both of them. In order not to overload notation, we will therefore abstain from making this difference explicit—except for Theorem 5, where it is really crucial.

Using the ring $\mathfrak{An}(\mathfrak{F})$, the operator-theoretic formulation of (3) can now be written as a system of polynomial equations, but this implies that all the basic operators occurring as indeterminates are void of any analytic meaning. Therefore we have to add appropriate *interaction equalities* for algebraically capturing their essential properties. For example, the interaction between differentiation and multiplication operators is stated in the well-known Leibniz "equality". For other operator interactions, the corresponding equalities are less obvious, and completeness questions (confluence, termination, adequacy) become urgent.

For the moment, however, we postpone these issues to Subsection 2.5, which shows the full polynomial system along with the corresponding completeness theorems. So we assume we have got an appropriate reduction system, which we will now employ for solving the given polynomial system $T\,G = 1$ and $B_1\,G = \cdots = B_n\,G = 0$. In principle, we could merge these equations with the interaction equalities, impose a lexicographic term order, and feed the whole thing into a noncommutative Gröbner basis solver; this is essentially what we have done in (Rosenkranz, Buchberger, Engl, 2003). However, we can do better than that, using a generic *preprocessing strategy* that avoids computing a Gröbner basis for each new BVP of type (3); see Subsection 2.5.

2.2 The Moore-Penrose Inverse

The key to simplifying the given polynomial system is some basic *Moore-Penrose theory* (Nashed, 1976; Engl, Hanke, Neubauer, 1996). Well-known from the finite-dimensional case ("generalized matrix inverse"), this method provides a substitute for inverting a non-bijective linear operator in any vector space—including the space $C^\infty[a, b]$ used by us.

Why would we want to do this? For a linear differential operator T, we have to solve $T\,G = 1$ for G, subject to the additional conditions $B_1\,G = \cdots = B_n\,G = 0$, which are supposed to determine the solution uniquely. So we are searching a special right inverse G of T. The usual way of seeing this is that G is the *full inverse* (not just right inverse) of the operator T by restricting the domain of the latter to those functions in $C^\infty[a, b]$ that fulfill the given boundary conditions.

Though theoretically elegant, this interpretation is not adequate for our purposes as it encodes the boundary conditions in the domain definition, where it is not easily available for computations. Therefore we change our perspective by seeing the given operator T as non-bijective, having all of $C^\infty[a,b]$ as its domain just like the other basic operators. In this case, we can employ the Moore-Penrose theory for finding *generalized inverses* of T, and we have to find some means of selecting the appropriate one among all possible choices by incorporating the boundary conditions.

This can be achieved by using *oblique Moore-Penrose inverses* (Nashed, 1976, pages 57–61). The idea is the following: An arbitrary linear operator T between two vector spaces X and Y may fail to be injective, so its nullspace N is typically nontrivial. In order to cure this, one takes a complement M by choosing a projector P onto N and setting $M = (1-P)X$; the operator T restricted to M is then invertible as a map from X to its range R. Furthermore, T may fail to be surjective, so R will typically not exhaust all of Y. For repairing this, one chooses a projector Q onto R, calls the corresponding complement $S = (1-Q)Y$, and extends $(T|_M)^{-1}$ with nullspace S; the resulting operator is called the oblique Moore-Penrose inverse $T^\dagger_{P,Q}$ of T with respect to the chosen nullspace projector P and range projector Q. The freedom in choosing these projectors will be crucial for incorporating the boundary conditions.

What makes the Moore-Penrose inverse particularly attractive for symbolic computation is that it can be uniquely characterized by the four so-called *Moore-Penrose equations.* Let us briefly recall them here for reference purposes.

Theorem 1 *Let X and Y be vector spaces, T a linear operator from X to Y. Choose projectors P and Q to the nullspace and range of T, respectively, and let M and S be the corresponding complements. Then the oblique Moore-Penrose inverse is uniquely characterized as a linear operator $T^\dagger$ from Y to X fulfilling the equations*

$$T\,T^\dagger T = T, \tag{5}$$
$$T^\dagger T T^\dagger = T^\dagger, \tag{6}$$
$$T^\dagger T = 1 - P \tag{7}$$
$$T\,T^\dagger = Q. \tag{8}$$

Furthermore, $T^\dagger$ has nullspace S and range M.

It is already clear that Q must be the identity operator 1, because any linear differential operator is surjective on $C^\infty[a,b]$. But then (5) and (6) obviously follow from (8). So we are left with the two equations (7) and (8). It turns out, however, that we can even restrict ourselves to (7) because (8) follows from it as we will show now.

Lemma 1 *The operator equation $T\,G = 1$ follows from $G\,T = 1 - P$, where P is some nullspace projector for the linear differential operator T.*

Proof. Let T^* be any right inverse of T (there is always a right inverse or—in other words—a fundamental solution for T, and we will construct a particular one in Subsection 2.4). Then premultiplying $G\,T = 1 - P$ by T and postmultiplying by T^* yields $TGTT^* = TT^* - TPT^*$. Now by the choice of T^*, we have $TT^* = 1$, and since P projects onto the nullspace, we have $TP = 0$. Hence $TG = 1$ as claimed. □

As a consequence, we need only consider the equation $G\,T = 1 - P$, but we must take care to choose P in such a way that the *boundary conditions* $B_1\,G = \cdots = B_n\,G = 0$ are also respected. Then we can be sure that G is actually the Green's operator: Since it is uniquely determined, it must coincide with the single Moore-Penrose inverse of T corresponding to that choice of P that incorporates the boundary conditions.

2.3 Computation of the Nullspace Projector

For that purpose, we use the fact mentioned at the end of Theorem 1, namely that the range of G is given by

$$(1 - P)\,C^\infty[a,b] = \{v - P\,v | v \in C^\infty[a,b]\}.$$

So if we want to ensure that the solution $u = G\,f$ respects the boundary conditions $B_1\,u = \cdots = B_n\,u = 0$ for any $f \in C^\infty[a,b]$, it suffices to construct P in such a way that all the $v - P\,v$ respect them, so we must require

$$\begin{aligned} &B_1 P\,v = B_1\,v \\ &\cdots \\ &B_n P\,v = B_n\,v \end{aligned} \tag{9}$$

for all $v \in C^\infty[a,b]$. This amounts to an easy task of *linear interpolation*, which is solved by the next lemma.

For that purpose, let us introduce some *matrix notation* (we will use overhat symbols for denoting vectors and matrices). We write $\hat{D}_n$ for the operator-valued vector $(1, D, D^2, \ldots, D^{n-1})$. With this notation, the vector boundary operator $\hat{B} = (B_1, \ldots, B_n)$ can be written as $(L\,\hat{l} + R\,\hat{r})\,\hat{D}_n$ for suitable $\hat{l}, \hat{r} \in \mathbb{R}^{n\times n}$. In fact, using the notation of (2), these matrices are given by

$$\hat{l} = \begin{pmatrix} p_{1,0} & p_{1,1} & \cdots & p_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,0} & p_{n,1} & \cdots & p_{n,n-1} \end{pmatrix}, \qquad \hat{r} = \begin{pmatrix} q_{1,0} & q_{1,1} & \cdots & q_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n,0} & q_{n,1} & \cdots & q_{n,n-1} \end{pmatrix}.$$

Now we are ready to state a neat *formula for computing the nullspace projector* in terms of $\hat{l}, \hat{r}$ and the fundamental matrix associated with the given differential operator T.

Lemma 2 *Let $\hat{w}$ be the fundamental matrix of T and let $\hat{l}, \hat{r}$ be the boundary matrices corresponding to $B_1, \ldots, B_n$ as introduced above. Compute*

$$\operatorname{Proj}_{\hat{w}}(\hat{l}, \hat{r}) = \lceil \hat{w}_1 \rceil \, (\hat{l}\hat{w}^{\leftarrow} + \hat{r}\hat{w}^{\rightarrow})^{-1}(L\,\hat{l} + R\,\hat{r})\,\hat{D}_n,$$

where $\hat{w}_1$ denotes the first row of $\hat{w}$ and $\hat{w}^{\leftarrow}$ and $\hat{w}^{\rightarrow}$ arise from $\hat{w}$ by evaluation at a and b, respectively. Then $\operatorname{Proj}_{\hat{w}}(\hat{l}, \hat{r})$ is a projector onto the nullspace of T fulfilling (9).

Proof. Let T be an operator of the form (1) and let $B_1, \ldots, B_n$ be boundary operators of the form (2) with corresponding boundary matrices $\hat{l}, \hat{r}$. Furthermore, let $\varphi_1, \ldots, \varphi_n$ be a fundamental system for T; hence the fundamental matrix $\hat{w}$ has rows $(\varphi_1^{(i)}, \ldots, \varphi_n^{(i)})$ for $i = 0, \ldots, n-1$.

We will now set up a generic linear operator P that projects onto the nullspace of T and fit it against the conditions of (9). Take an arbitrary $v \in C^\infty[a,b]$. Since the nullspace of T is spanned by $\varphi_1, \ldots, \varphi_n$, we must have $P\,v = c_1(v)\,\varphi_1 + \ldots + c_n(v)\,\varphi_n$ for some coefficients $c_1, \ldots, c_n \in \mathbb{C}$ depending on v. Writing this in vector form, we have $P\,v = \hat{w}_1\,\hat{c}(v)$, which yields the matrix equation $\hat{B}\,\hat{w}_1\,\hat{c}(v) = \hat{B}\,v$ upon substitution in (9). Now

$$\hat{B}\,\hat{w}_1 = (L\,\hat{l} + R\,\hat{r})\,\hat{D}_n\,\hat{w}_1 = (L\,\hat{l} + R\,\hat{r})\,\hat{w} = \hat{l}\hat{w}^{\leftarrow} + \hat{r}\hat{w}^{\rightarrow},$$

so $\hat{c}(v) = (\hat{l}\hat{w}^{\leftarrow} + \hat{r}\hat{w}^{\rightarrow})^{-1}\,\hat{B}\,v$, which yields $P = \operatorname{Proj}_{\hat{w}}(\hat{l}, \hat{r})$ as claimed in the lemma. □

Note that that the *matrix inversion* occurring in the Lemma 2 applies only to a matrix with numerical constants rather than functional terms.

2.4 Right Inversion

We have now reduced the BVP (3) to the single equation $G\,T = 1 - P$, where P is the nullspace projector $\operatorname{Proj}_{\hat{w}}(\hat{l}, \hat{r})$ as defined in Lemma 2 with $\hat{w}$ the fundamental matrix for T and $\hat{l}, \hat{r}$ the boundary matrices corresponding to $B_1, \ldots, B_n$. In order to solve this equation for G, it suffices to find a *right inverse* T^* of T; then G can be written as $(1-P)\,T^*$. We will construct one particular such right inverse of T, which we will denote by $T^\blacklozenge$.

It turns out that one can always find right inverses of T that can be written in a form analogous to (4) with a binary function g^*; in the literature (Kamke, 1983, page 74), this function is known as the *fundamental solution* of the inhomogeneous differential equation $T\,u = f$. The fundamental solution plays a role somewhat similar to the Green's function: When applying the corresponding integral operator to the forcing function f, it yields a solution u of the inhomogeneous equation, but it does not incorporate boundary conditions.

In (Rosenkranz, 2003), we have only treated the simple but important case of linear differential operators with *constant coefficients.* It turns out that for such operators, there is also a particularly simple formula for right inversion—whereas there seems to be no significant advantage when applying the procedure from (Kamke, 1983) to linear differential operators with constant coefficients.

Lemma 3 *If T is of the form* (1) *with constant coefficient functions $c_0, \ldots, c_n$, the operator*

$$T^{\blacklozenge} = \prod_{i=1}^{n} \lceil e^{\lambda_i x} \rceil \, A \, \lceil e^{-\lambda_i x} \rceil$$

is a right inverse, where $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$ are the roots of the characteristic polynomial of T (repeated according to their multiplicities).

Proof. For arbitrary $\lambda \in \mathbb{C}$, the differential operator $D - \lambda$ has $\lceil e^{\lambda x} \rceil \, A \, \lceil e^{-\lambda x} \rceil$ as a right inverse as one can see by straightforward computation, using the product rule of differentiation and the fundamental theorem of calculus (see Subsection 2.5 for a precise listing of admissible reduction rules). The formula then follows since

$$T = (D - \lambda_1) \cdots (D - \lambda_n)$$

and operator composition is associative. □

The general case of linear differential operators with *variable coefficients* is slightly more complicated. But it turns out that right inversion is essentially an iteration of a procedure typically called "reduction of order" in the literature (Coddington, Levinson, 1955, page 84).

Lemma 4 *Let T be a linear differential operator of the form* (1) *having fundamental system $\{\varphi_1, \ldots, \varphi_n\}$. Now define the following triangular system of functions $\langle \varphi_{jk} \,|\, j = 1, \ldots, n \wedge k = j, \ldots, n \rangle$ by recursion. Set $\varphi_{1k} = \varphi_k$ for all $k = 1, \ldots, n$, and $\varphi_{j+1,k} = (\varphi_{jk}/\varphi_{jj})'$ for all $j = 1, \ldots, n-1$ and $k = j+1, \ldots, n$. Then*

$$T^{\blacklozenge} = \Phi \lceil \varphi^{-1} \rceil \quad \text{with} \quad \Phi = \prod_{k=1}^{n} (\lceil \varphi_{kk} \rceil \, A) \quad \text{and} \quad \varphi = c_n \prod_{k=1}^{n} \varphi_{kk}$$

is a right inverse for T.

Proof. The differential operator $T \lceil \varphi_1 \rceil$ annihilates 1 and hence has vanishing constant coefficient. Then $T \lceil \varphi_1 \rceil A$ is a differential operator of order $n - 1$

with fundamental system $\{(\varphi_2/\varphi_1)', \dots, (\varphi_n/\varphi_1)'\}$. Iterating this procedure, one obtains the zero-order differential operator $T\Phi$ with Φ defined as specified. Hence $T\Phi = \lceil\varphi\rceil$ for some function φ yet to be determined. Now on the one hand, we have

$$T\Phi = (\lceil c_n\rceil D^n + \cdots + \lceil c_1\rceil D + \lceil c_0\rceil)\,\lceil\varphi_{11}\rceil A\lceil\varphi_{22}\rceil A\cdots\lceil\varphi_{nn}\rceil A,$$

which can be expanded and simplified (see the reduction rules listed in Subsection 2.5 to make this precise) to an analytic polynomial with monomials containing—apart from multiplication operators— only D^k for $k = 0, \dots, n$ or A. But on the other hand, this polynomial coincides with $\lceil\varphi\rceil$, so the only monomial left over is the one with D^0. And the only contribution of this kind comes from $\lceil c_n\rceil D^n\,\Phi$ by applying the product rule n times, always keeping the second monomial of the result (see the middle equality for isolating differential operators). Thus one indeed ends up with φ as stated in the lemma. □

Note that we must perform division for the case of variable coefficents, so the analytic algebra over which we work must now be a *field algebra* (see the end of Subsection 2.5). A typical example is the algebra of all "rational functions in exponentials", namely $\mathbb{Q}(x, (e^{\lambda x})_{\lambda\in\mathbb{C}^*})$; basis expansion in this algebra is essentially partial fraction decomposition.

2.5 The Reduction System

Using the above results, we can compute the desired Green's operator G as $(1 - P)\,T^\blacklozenge$, where P is again $\mathrm{Proj}_{\hat{w}}(\hat{l}, \hat{r})$ as in Lemma 2 and $T^\blacklozenge$ is the right inverse specified in Lemma 3 or Lemma 4. However, we might obtain G in a somewhat *unconventional form*: For example, in the BVP for the heat equation (see Subsection 3.1), we have $T = D^2$ and $B_1 = L, B_2 = R$. In this case, Lemma 2 yields $P = \lceil 1 - x\rceil\, L + \lceil x\rceil\, R$, while Lemma 3 gives us of course $T^\blacklozenge = A^2$. Hence we have $G = (1 - \lceil 1 - x\rceil\, L + \lceil x\rceil\, R)\, A^2$.

This Green's operator is not of the classical kernel representation (4), and we cannot read off the Green's function g associated with it. Using the obvious simplification $L\,A = 0$, we can also rewrite G into $A^2 - \lceil x\rceil A^2$. The representation via the Green's function in Subsection 3.1 is a third possibility. In general, there are many different polynomials in $\mathfrak{An}(\mathfrak{Exp})$ with the same interpretation as an operator on $C^\infty[a, b]$. Our goal is to organize rewriting in such a way that there will always be a *unique* final result, which will moreover correspond to the classical kernel representation.

But before doing so, we would like to point out that the issue of representations is actually peripheral to the original problem of solving a BVP of the form (3):

Whatever representation of G we take, when we apply it to a given forcing function f, we will end up with the unique solution $u = G\,f$ of the BVP—as long as the reduction system is *sound* in the sense to be discussed now.

In order to realize our goal, we have to set up an appropriate reduction system on the ring of analytic polynomials. As usual the reductions are specified for a set of monomials and are to be extended in the obvious way—see for example (Bergman, 1979). The reduction system should have the following five *key*

Equalities for Algebraic Simplification:	*Equalities for Contracting Integration Operators:*
$\lceil f \rceil \lceil g \rceil \to \lceil f g \rceil$	$A \lceil f \rceil A \to \lceil \int^* f \rceil A - A \lceil \int^* f \rceil$
Equalities for Isolating Differential Operators:	$A \lceil f \rceil B \to \lceil \int^* f \rceil B + A \lceil \int^* f \rceil$
	$B \lceil f \rceil A \to \lceil \int_* f \rceil A + B \lceil \int_* f \rceil$
$D\,A \to 1$	$B \lceil f \rceil B \to \lceil \int_* f \rceil B - B \lceil \int_* f \rceil$
$D\,B \to -1$	$A\,A \to \lceil \int^* 1 \rceil A - A \lceil \int^* 1 \rceil$
$D \lceil f \rceil \to \lceil f \rceil D + \lceil f' \rceil$	$A\,B \to \lceil \int^* 1 \rceil B + A \lceil \int^* 1 \rceil$
$D\,L \to 0$	$B\,A \to \lceil \int_* 1 \rceil A + B \lceil \int_* 1 \rceil$
$D\,R \to 0$	$B\,B \to \lceil \int_* 1 \rceil B - B \lceil \int_* 1 \rceil$
Equalities for Isolating Boundary Operators:	*Equalities for Absorbing Integration Operators:*
$L\,A \to 0$	$A \lceil f \rceil D \to -f^{\leftarrow} L + \lceil f \rceil - A \lceil f' \rceil$
$R\,A \to A + B$	$B \lceil f \rceil D \to f^{\rightarrow} R - \lceil f \rceil - B \lceil f' \rceil$
$L\,B \to A + B$	$A\,D \to -L + 1$
$R\,B \to 0$	$B\,D \to R - 1$
$L \lceil f \rceil \to f^{\leftarrow} L$	$A \lceil f \rceil L \to \lceil \int^* f \rceil L$
$R \lceil f \rceil \to f^{\rightarrow} R$	$B \lceil f \rceil L \to \lceil \int_* f \rceil L$
$L\,L \to L$	$A \lceil f \rceil R \to \lceil \int^* f \rceil R$
$L\,R \to R$	$B \lceil f \rceil R \to \lceil \int_* f \rceil R$
$R\,L \to L$	$A\,L \to \lceil \int^* 1 \rceil L$
$R\,R \to R$	$B\,L \to \lceil \int_* 1 \rceil L$
Table 1:	$A\,R \to \lceil \int^* 1 \rceil R$
THE GREEN'S SYSTEM	$B\,R \to \lceil \int_* 1 \rceil R$

properties:

- It must be *sound* in the sense that each polynomial equation becomes a valid identity of operators when interpreted in the obvious way.
- It must be *adequate* in the sense that it provides “enough” reductions for algebraizing all the analytic knowledge relevant here.
- In order to solve the problem of unique representation addressed above, we require it to be *confluent*: there is no more than one normal form.
- Besides this, every simplification should terminate, i.e. the reduction system must be *noetherian*: there is at least one normal form.
- The normal forms of the reduction system should correspond exactly to the Green's functions of the classical kernel representation (4). Hence we will also refer to these normal forms as *Green's polynomials.*

The reduction system in Table 1—we have called it the Green's system—fulfills all these requirements. For a complete *proof* of this statement, see (Rosenkranz, 2003); here we will only point out a few basic features of the proof.

First of all, let us clarify the role of the *analytic algebra* $\mathfrak{F}$ already mentioned in Definition 1; the variables f and g in Table 1 range over its basis $\mathfrak{F}^{\#}$. Analytic algebras are simply algebras with a few additional operations fufilling certain axioms that make them behave similar to their analytic models—just like differential algebras, which can be seen as halfway between plain algebras and analytic algebras.

Definition 2 *An algebra $\mathfrak{F}$ is called an* analytic algebra *iff it has five linear operations: differentation $'$: $\mathfrak{F} \to \mathfrak{F}$, integral $\int^*$: $\mathfrak{F} \to \mathfrak{F}$, cointegral $\int_*$: $\mathfrak{F} \to \mathfrak{F}$, left boundary value $^{\leftarrow}$: $\mathfrak{F} \to \mathbb{C}$, right boundary value $^{\rightarrow}$: $\mathfrak{F} \to \mathbb{C}$ such that the seven axioms*

$$\begin{aligned}
(fg)' &= f'g + fg',\\
\textstyle\int^* f' &= f - f^{\leftarrow},\\
\textstyle\int_* f' &= f^{\rightarrow} - f,\\
(\textstyle\int^* f)' &= f,\\
(\textstyle\int_* f)' &= -f,\\
(fg)^{\leftarrow} &= f^{\leftarrow} g^{\leftarrow},\\
(fg)^{\rightarrow} &= f^{\rightarrow} g^{\rightarrow}.
\end{aligned}$$

are fulfilled.

We observe that the above *axioms* are very natural: The first is the product rule for differentiation, thus making analytic algebras a special case of differential algebras (where this axiom is usually called the Leibniz rule). The next four axioms state that the integral and the negative cointegral are oblique Moore-Penrose inverses of differentiation, having as nullspace projectors the left and right boundary value, respectively (with trivial range projectors in both cases); cf. the Moore-Penrose equations in Theorem 1. So the operations $^{\leftarrow}$ and $^{\rightarrow}$ serve to choose among the oblique Moore-Penrose inverses by fixing the integration constant. The last two axioms stipulate that $f \mapsto (x \mapsto f^{\leftarrow})$ and $f \mapsto (x \mapsto f^{\rightarrow})$ be homomorphisms in the algebra $\mathfrak{F}$.

As mentioned before, a typical choice for $\mathfrak{F}$ is the polyexponentials $\mathfrak{Exp}$. It can easily be verified that they form indeed an analytic algebra. Of course its operations will in general transform basis elements to non-basis elements; for example, $x\,e^x \in \mathfrak{Exp}^\#$ becomes $e^x + xe^x \in \mathfrak{Exp} \setminus \mathfrak{Exp}^\#$ under differentiation. So strictly speaking, the right-hand sides of Table 1 will not constitute polynomials from $\mathfrak{An}(\mathfrak{F})$ anymore, because the latter may only involve multiplication operators induced by basis elements. Therefore the reduction rules are to be understood as containing an implicit *basis reduction* after applying them: Any occurrence of a monomial $\cdots \lceil f \rceil \cdots$ with $f \in \mathfrak{F} \setminus \mathfrak{F}^\#$ is to be replaced by $\sum c_i \cdots \lceil f_i \rceil \cdots$, where $\sum c_i f_i$ is the basis expansion of f with non-zero coefficients $c_i \in \mathbb{C}$ and basis functions $f_i \in \mathfrak{F}^\#$.

The axioms for analytic algebras play a crucial role in *establishing the confluence* of the Green's system. What we have actually proved is that for every analytic algebra $\mathfrak{F}$, the system of Table 1 establishes a confluent reduction on the ring of analytic polynomials $\mathfrak{An}(\mathfrak{F})$. It is enough to consider the case $\mathfrak{F}^\# = \mathfrak{F}$, as one can easily see. By Lemma 1.2 of (Bergman, 1979), it suffices to prove that all overlap ambiguities of the reduction system are resolvable (in general, one also has to consider inclusion ambiguities, but by inspecting Table 1 we see that there are no inclusions in our case). We do this in the usual manner by showing that the S-polynomial $w_2 p_1 - p_2 w_1$ reduces to 0 for any pair of rules $ww_1 \to p_1$ and $w_2 w \to p_2$.

It turns out that there are 233 S-polynomials to be considered, so the task of doing all these reductions is rather daunting. It is therefore preferrable to *automate this proof*. As we have implemented the whole algorithm for computing Green's operators in the TH∃OREM∀ system (see Subsection 2.6 for some details), it seems natural to do this also in TH∃OREM∀—a neat example of how this system offers support on various levels: Here, on the object level of computation (using the reduction system for computing as explained below) as well as on the meta level of proof (verifying properties of the system, like confluence in our case).

For doing so, we have hand-proved some auxiliary equalities valid in any analytic algebra $\mathfrak{F}$. These equalities are mainly integral theorems like

$$\int^* \left(f\,(\textstyle\int^* f)\right) = \frac{1}{2}\left(\int^* f\right)^2;$$

see (Rosenkranz, 2003) for details. Tables 2 and 3 show a small fragment of the

The rules DA and AMA yield the S-polynomial:

$$\lceil f\rceil A - D\left\lceil\int^* f\right\rceil A + D\,A\left\lceil\int^* f\right\rceil \overset{\overset{\text{(...)}}{\downarrow}}{=}$$

$$\lceil f\rceil A - D\left\lceil\int^* f\right\rceil A + \boxed{D\,A}\left\lceil\int^* f\right\rceil \overset{\overset{\text{(DA)}}{\downarrow}}{=}$$

$$\left\lceil\int^* f\right\rceil + \lceil f\rceil A - \boxed{D\left\lceil\int^* f\right\rceil}A \overset{\overset{\text{(DM)}}{\downarrow}}{=}$$

$$\left\lceil\int^* f\right\rceil + \lceil f\rceil A - \left\lceil\boxed{\left(\int^* f\right)'}\right\rceil A - \left\lceil\int^* f\right\rceil D\,A \overset{\overset{\text{(da)}}{\downarrow}}{=}$$

$$\left\lceil\int^* f\right\rceil - \left\lceil\int^* f\right\rceil\boxed{D\,A} \overset{\overset{\text{(DA)}}{\downarrow}}{=}$$

$0 \qquad \square$

…

The rules RA and AMA yield the S-polynomial:

$$A\lceil f\rceil A + B\lceil f\rceil A - R\left\lceil\int^* f\right\rceil A + R\,A\left\lceil\int^* f\right\rceil \overset{\overset{\text{(...)}}{\downarrow}}{=}$$

$$A\lceil f\rceil A + B\lceil f\rceil A - R\left\lceil\int^* f\right\rceil A + \boxed{R\,A}\left\lceil\int^* f\right\rceil \overset{\overset{\text{(RA)}}{\downarrow}}{=}$$

$$A\left\lceil\int^* f\right\rceil + B\left\lceil\int^* f\right\rceil + A\lceil f\rceil A + B\lceil f\rceil A - \boxed{R\left\lceil\int^* f\right\rceil}A \overset{\overset{\text{(RM)}}{\downarrow}}{=}$$

$$A\left\lceil\int^* f\right\rceil + B\left\lceil\int^* f\right\rceil - \boxed{\left(\int^* f\right)^{\rightarrow}}R\,A + A\lceil f\rceil A + B\lceil f\rceil A \overset{\overset{\text{(ra)}}{\downarrow}}{=}$$

$$A\left\lceil\int^* f\right\rceil + B\left\lceil\int^* f\right\rceil - \left(\oint f\right)\boxed{R\,A} + A\lceil f\rceil A + B\lceil f\rceil A \overset{\overset{\text{(RA)}}{\downarrow}}{=}$$

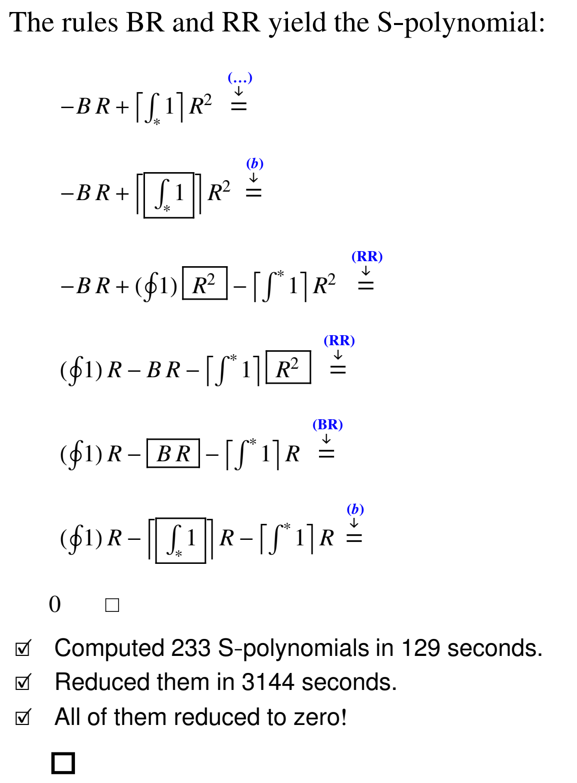
$$-\left(\oint f\right)A - \left(\oint f\right)B + A\left\lceil\int^* f\right\rceil + B\left\lceil\int^* f\right\rceil + \boxed{A\lceil f\rceil A} + B\lceil f\rceil A \quad ^{(}$$

$$-\left(\oint f\right)A - \left(\oint f\right)B + B\left\lceil\int^* f\right\rceil + \left\lceil\int^* f\right\rceil A + \boxed{B\lceil f\rceil A} \overset{\overset{\text{(BMA)}}{\downarrow}}{=}$$

$$-\left(\oint f\right)A - \left(\oint f\right)B + B\left\lceil\int^* f\right\rceil + B\left\lceil\boxed{\int_* f}\right\rceil + \left\lceil\int^* f\right\rceil A + \left\lceil\boxed{\int_* f}\right\rceil$$

$0 \qquad \square$

Table 2:
FRAGMENT OF THE CONFLUENCE PROOF



The rules BR and RR yield the S-polynomial:

$$-B\,R+\left\lceil \int_{*} 1\right\rceil R^2 \overset{\substack{(\ldots)\\ \downarrow}}{=}$$

$$-B\,R+\left\lceil \boxed{\int_{*} 1}\right\rceil R^2 \overset{\substack{(b)\\ \downarrow}}{=}$$

$$-B\,R+(\oint 1)\boxed{R^2}-\left\lceil \int^{*} 1\right\rceil R^2 \overset{\substack{(\mathrm{RR})\\ \downarrow}}{=}$$

$$(\oint 1)\,R-B\,R-\left\lceil \int^{*} 1\right\rceil \boxed{R^2} \overset{\substack{(\mathrm{RR})\\ \downarrow}}{=}$$

$$(\oint 1)\,R-\boxed{B\,R}-\left\lceil \int^{*} 1\right\rceil R \overset{\substack{(\mathrm{BR})\\ \downarrow}}{=}$$

$$(\oint 1)\,R-\left\lceil \boxed{\int_{*} 1}\right\rceil R-\left\lceil \int^{*} 1\right\rceil R \overset{\substack{(b)\\ \downarrow}}{=}$$

$0 \quad \square$

☑ Computed 233 S-polynomials in 129 seconds.

☑ Reduced them in 3144 seconds.

☑ All of them reduced to zero!

□

Table 3:
FRAGMENT OF THE CONFLUENCE PROOF (CONT'D)

actual confluence proof (everything in these tables is verbatim TH∃OREM∀ output), which covers approximately 2000 lines altogether. In every intermediate expression, the redex is framed by the system in order to improve readability. The uppercase letters above the equality symbol refer to the corresponding rules of Table 1 (the names are derived from the monomial on the left-hand side, with multiplication operators generically denoted by the letter M); the lowercase letters refer to the auxiliary equalities. The expression $\oint f$, with $f \in \mathfrak{F}$, is an abbreviation for the "definite integral" $\int^* f + \int_* f$.

For establishing the *termination* of the Green's system, we have given two different proofs in (Rosenkranz, 2003). The more intuitive proof uses the idea of various termination terms associated with the rules. For example, several rules decrease the "differential weight" (the number of occurrences of the indeterminate D), whereas none of the rules increases it. The other proof goes on a more algebraic line: We set up a suitable graded lexicographic ordering on the word monoid Ω^* over $\Omega = \{D, A, B, L, R, M\}$, which is then extended to a well-ordering on the system of finite subsets of Ω^*. This well-ordering induces a noetherian strict partial order on $\mathfrak{An}(\mathfrak{F})$ by identifying all $\lceil f \rceil$ with M and taking the support of the resulting polynomial. Hence it suffices to prove that the reductions are compatible with the induced order, and this is easily achieved.

Summarizing the previous two results, we have proved convergence (i.e. confluence and termination) for the Green's system.

Theorem 2 *For any analytic algebra $\mathfrak{F}$, the system specified in Table 1 constitutes a convergent rewrite system on the ring of analytic polynomials $\mathfrak{An}(\mathfrak{F})$.*

As mentioned before, we can also characterize the normal forms (which always exist and are unique by the preceding theorem), which will turn out to be precise analogs of the Green's functions.

Definition 3 *A polynomial of $\mathfrak{An}(\mathfrak{F})$ is said to be a* Green's polynomial *iff all its monomials are produced by the rule $\mathcal{M}$ of the grammar in Table 4.*

Production Rule	*Name*
$\mathcal{M} ::= \mathcal{A}\mathcal{I}\mathcal{A} \mid \mathcal{A}\mathcal{D} \mid \mathcal{A}\mathcal{B}\mathcal{D}$	Monomial Operator
$\mathcal{I} ::= A \mid B$	Integral Operator
$\mathcal{A} ::= 1 \mid \lceil f \rceil$	Algebraic Operator
$\mathcal{B} ::= L \mid R$	Boundary Operator
$\mathcal{D} ::= 1 \mid D\mathcal{D}$	Differential Operator

Table 4:
GRAMMAR OF GREEN'S POLYNOMIALS

We denote the set of Green's polynomials by $\mathfrak{Gr}_{\downarrow}(\mathfrak{F})$.

Theorem 3 *The normal forms of $\mathfrak{An}(\mathfrak{F})$ with respect to the reduction system specified in Table 1 are precisely the Green's polynomials $\mathfrak{Gr}_{\downarrow}(\mathfrak{F})$.*

The proof of the preceding theorem is rather straight-forward, albeit slightly technical. It is easy to see that any Green's polynomial is indeed irreducible. For proving the converse, one takes an arbitrary monomial $p \in \mathfrak{An}(\mathfrak{F}) \backslash \mathfrak{Gr}_{\downarrow}(\mathfrak{F})$ and shows that it is reducible, using a case distinction on the first letters of p. Despite its rather technical proof, the statement of the theorem is actually *very intuitive*: Any linear integro-differential-boundary operator must be a superposition of purely integral or differential or boundary operators (algebraic operators can be seen as zero-order differential operators). This is clear because on the "atomic" (viz. monomial) level, integration and differentiation cancel each other, whereas boundary evaluation collapses the functional range to a single point.

It is now easy to see why a Green's polynomial allows to read off the corresponding *Green's function*. Since we know that the "differential weight" is invariant under the Green's system, the normal form of a Green's operator cannot be of type $\mathcal{AD}$ or $\mathcal{ABD}$; hence it must be of type $\mathcal{AIA}$. So each monomial has the form $\lceil f \rceil A \lceil g \rceil$ or $\lceil f \rceil B \lceil g \rceil$ (where f or g may also be 1), thus contributing the term $f(x)\,g(\xi)$ to the "upper" or "lower" part of the Green's function, which is defined by the case distinction

$$g(x,\xi) = \begin{cases} \text{upper}(x,\xi) \text{ if } a \le \xi \le x \le b, \\ \text{lower}(x,\xi) \;\text{ if } a \le x \le \xi \le b, \end{cases}$$

reflecting the characteristic jump on the diagonal of $[a,b] \times [a,b]$.

By a completely analogous process one can also extract a binary function h from the right inverse $T^{\blacklozenge}$ of the given differential operator T just as one extracts the Green's function g from the corresponding Green's operator G. In the literature, the function h is known as the *fundamental solution* of the differential equation $Tu = f$. Its role is similar to g, only that it ignores boundary conditions: For any forcing function f, the convolution defined by (4), with h instead of g, yields *some* solution u of the differential equation $Tu = f$. Comparing this with the relation $G = (1-P)\,T^{\blacklozenge}$, we gain a new interpretation of the fundamental solution: It is the "Green's function" associated with the trivial nullspace projector $P = 0$ (which can of course never arise from given boundary conditions).

Before clarifying the relations between the actual operators acting on $C^\infty[a,b]$ and their formal counterparts in the algebraic structure $\mathfrak{An}(\mathfrak{F})$, let us investigate the latter a bit more. For this purpose, we will now view the results about the reduction system in Table 1 from a *ring-theoretic perspective.*

Definition 4 *Let $\mathfrak{F}$ be an analytic algebra. Then $\mathfrak{Gr}_0(\mathfrak{F})$ denotes the* Green's system*, i.e. the set of all polynomials $l - r$ where $l \to r$ is a rule of the reduction system in Table 1 (with the variables f, g ranging over all of $\mathfrak{F}^\#$). Furthermore, $\mathfrak{Gr}(\mathfrak{F})$ denotes the two-sided ideal generated by $\mathfrak{Gr}_0(\mathfrak{F})$ in $\mathfrak{An}(\mathfrak{F})$; we call it the* Green's ideal *over $\mathfrak{F}$.*

Theorem 4 *For any analytic algebra $\mathfrak{F}$, the Green's system $\mathfrak{Gr}_0(\mathfrak{F})$ constitutes a noncommutative Gröbner basis for the ideal $\mathfrak{Gr}(\mathfrak{F})$ in $\mathfrak{An}(\mathfrak{F})$.*

The notion of *Gröbner bases* was originally introduced in the "classical" context of commutative polynomials by the second author in his PhD thesis Buchberger (1965); see also the journal version Buchberger (1970) and a consice treatment in Buchberger (1998). As discovered by Mora (1986, 1988), the computation of Gröbner bases can be transferred to noncommutative rings in a straight-forward way (though it may not terminate in all cases). Actually, there are several variations on the notion of noncommuative Gröbner bases; our usage is in harmony with Theorem 8 of Ufnarovski (1998). In our context, the essential idea of Gröbner bases is the confluence of the induced reduction—as we have already seen before, without using ring-theoretic terminology.

This leads us back to our remarks at the close of Subsection 2.1: It is now clear why we can avoid the costly computation of per-problem Gröbner bases as in (Rosenkranz, Buchberger, Engl, 2003): We already *have* $\mathfrak{Gr}_0(\mathfrak{F})$ as a Gröbner basis, and it need not be changed for the different instances of BVPs considered. Of course, $\mathfrak{Gr}_0(\mathfrak{F})$ is not a finite Gröbner basis since the variables f and g in Table 1 range over all functions in $\mathfrak{F}^\#$; however, it is *finitary* in the sense that it can be described by finitely many parametrized polynomials.

Finally we can now address the questions of soundness and adquacy—how the formal operators are related to the actual ones. For this, let us first clarify the *correspondence between analytic polynomials and actual operators* acting on $C^\infty[a,b]$.

Definition 5 *Let $\mathfrak{F}$ be an analytic algebra, $\mathfrak{A}$ an algebra containing $\mathfrak{F}$, and $\mathfrak{L}$ a subalgebra of the algebra of all linear operators on $\mathfrak{A}$. A homomorphism $I : \mathfrak{An}(\mathfrak{F}) \to \mathfrak{L}$ will be called an* interpretation of $\mathfrak{An}(\mathfrak{F})$ in $\mathfrak{L}$ *if $I(\lceil f \rceil)(a) = fa$ for all $f \in \mathfrak{F}$ and $a \in \mathfrak{A}$. It is called* sound *if all the equalities of Table 1 (where $\to$ is now regarded as $=$) are preserved.*

If $\mathfrak{L}$ is the algebra of all linear operators on the algebra of smooth functions $C^\infty[a,b]$, we define the *smooth interpretation* sm of $\mathfrak{An}(\mathfrak{F})$ in $\mathfrak{L}$ by setting

$$\begin{aligned}
\mathrm{sm}(D)(u) &= u', \\
\mathrm{sm}(A)(u) &= x \mapsto \textstyle\int_a^x u(\xi)\, d\xi, \\
\mathrm{sm}(B)(u) &= x \mapsto \textstyle\int_x^b u(\xi)\, d\xi, \\
\mathrm{sm}(L)(u) &= x \mapsto u(a), \\
\mathrm{sm}(R)(u) &= x \mapsto u(b), \\
\mathrm{sm}(\lceil f \rceil)(u) &= fu,
\end{aligned}$$

where u ranges over $C^\infty[a,b]$, x over $[a,b]$, and f over $\mathfrak{F}$. It is easy to see that sm is indeed sound (actually the equalities of Table 1 are extracted from relations in $\mathfrak{L}$). In a similar fashion, one may also define a distributional interpretation by using the algebra of boundary-valued distributions $C_0^{-\infty}[a,b]$ instead of $C^\infty[a,b]$. In fact, all the statements formulated for the smooth interpretation carry over to the distributional case (including "strong" and "weak" solutions); see (Rosenkranz, 2003, page 45) for details.

Finally we arrive now at the summit of this treatise: the correctness statement for our method of computing the Green's operator, at the same time asserting the *adequacy* of the Green's system in Table 1. The interpretation of an analytic polynomial p will be denoted by $\underline{p}$ (note that all differential and boundary operators in the problems considered here can be written in this form).

Theorem 5 *Assume we have*

- *a BVP* (3) *on the real interval* $[a, b]$*, given by a differential operator* $\underline{T}$ *and boundary operators* $\underline{B_1}, \dots, \underline{B_n}$*,*
- *subject to the restrictions specified at the beginning of Subsection 2.1,*
- *and an analytic field algebra* $\mathfrak{F}$ *that contains the coefficient functions and the fundamental system of* $\underline{T}$ *(in the case of constant coefficients the analytic non-field algebra* $\mathfrak{Exp}$ *is sufficient).*

Now compute

- *the nullspace projector* P *according to Lemma 2,*
- *the right inverse* $T^{\blacklozenge}$ *of* T *as in Lemma 4,*
- *the normal form* $G \in \mathfrak{Gr}_{\downarrow}(\mathfrak{F})$ *of* $(1-P)\,T^{\blacklozenge}$ *with respect to the Green's system in Table 1,*

Then $\underline{G}$ *is the Green's operator of the given BVP, and* G *represents the corresponding Green's function* g *of* (4).

Proof. By Lemma 2, $\underline{P}$ is indeed a projector onto the nullspace of $\underline{T}$. Since $\underline{T}$ is always surjective, $\underline{1}$ is the only possible projector onto the range of $\underline{T}$. Now there is a unique oblique Moore-Penrose inverse of T having these projectors; we will write it as $\underline{G}$ for some $G \in \mathfrak{An}(\mathfrak{F})$ yet to be determined.

By Theorem 1, $\underline{G}$ is also determined uniquely by the four Moore-Penrose equations (5)–(8). As explained after Theorem 1, we can restrict ourselves to (7) and (8); finally, Lemma 1 reduces everything to (7), which reads $\underline{GT} = \underline{1 - P}$. Since $\underline{TT^{\blacklozenge}} = \underline{1}$ by Lemma 4, postmultiplying by $\underline{T^{\blacklozenge}}$ yields $\underline{G} = \underline{(1 - P)\,T^{\blacklozenge}}$. Hence we may choose the normal form of $(1 - P)\,T^{\blacklozenge}$ for G, and its interpretation $\underline{G}$ will be the desired Moore-Penrose inverse.

For any $f \in C^\infty[a, b]$, the image $u = \underline{G}\,f$ fulfills the given differential equation $\underline{T}\,u = f$ because of the fourth Moore-Penrose equation (8). The range of $\underline{G}$ is $\underline{1 - P}\,C^\infty[a, b]$ by Theorem 1, and every function contained in this range fulfills the given boundary conditions by Lemma 2. Hence $\underline{G}\,f$ fulfills the given BVP for any $f \in C^\infty[a, b]$, and $\underline{G}$ must coincide with the desired Green's operator due to the regularity assumption. Moreover, G represents the Green's function g since G is a Green's polynomial; see the discussion after Theorem 3. □

2.6 An Implementation

As mentioned before, we have implemented our method in TH∃OREM∀—a *mathematical software system* devloped at RISC under the supervision of the second author. Based on the computer algebra software Mathematica$^{\text{TM}}$, this system offers support for proving, computing and solving in various mathematical domains. Our implementation for Green's functions is a good example of the interplay between these three fundamental activities in mathematics: For *solving* a BVP, we *compute* the Green's operator by a reduction system that is *proved* confluent (see Subsection refssec:redsys for more details).

The *core machinery* for computing the Green's operator by our method is concerned with handling noncommutative polynomials—this is mainly addition, subtraction, multiplication, reduction to normal form. We have implemented these operations as a separate "basic evaluator" named **ReduceNoncommutativePolynomial**. Based on TH∃OREM∀, it benefits from the neat notation facilities available there: One may write the noncommutative polynomials exactly as one would on paper (e.g. denoting multiplication by juxtaposition rather than $**$ as in plain Mathematica$^{\text{TM}}$).

The basic evaluator for noncommutative polynomials is used for computing the nullspace projector as in Lemma 2, the right inverse as in 3, and finally the Green's function as in Theorem 5. (Currently, we do not support differential operators with variable coefficients, though it is straightforward to use Lemma 4 instead of Lemma 3 for computing the right inverse.) All these *applied operations* are implemented in another basic evaluater named **GreenEvaluator**. In the next section, we will show some computations carried out by this evaluator (note that all the input and output is printed verbatim).

3 Sample Computations

3.1 Heat Conduction

The following problem seems to be one of the classical examples that are most often used for introducing the concepts of ordinary linear BVPs (Stakgold, 1979, page 42). It can be interpreted as describing *one-dimentional steady heat conduction in a homogeneous rod.* In its functional formulation (after scaling everything to unity), it means solving

$$\begin{aligned} u'' &= f, \\ u(0) &= u(1) = 0 \end{aligned}$$

for $u \in C^\infty[0,1]$ with a given heat source $f \in C^\infty[0,1]$.

In this example, we have the differential operator $T = D^2$, so the nullspace is $\{\alpha x + \beta \mid \alpha, \beta \in \mathbb{C}\}$, and finding the *nullspace projector* P reduces to the following linear interpolation problem: Given a function $v \in C^\infty[0,1]$, find a linear function $P\,v$ that agrees with v at the grid points $0, 1$. In our case we can do this automatically:

`In[1]:=` $\text{Compute}[\text{Proj}_{\hat{w}}, \text{by} \to \text{GreenEvaluator},$
$\quad \text{using} \to \text{KnowledgeBase}[\text{"ClassicalHeatConduction"}]]$

`Out[1]=` $L - \lceil x \rceil L + \lceil x \rceil R$

The other crucial step is to find the *right inverse* $(D^2)^{\blacklozenge}$. Trivially, this is A^2 in our case, but this is not in normal form. Computing it by our system returns the normal form:

`In[2]:=` $\text{Compute}[(D^2)^{\blacklozenge}, \text{by} \to \text{GreenEvaluator},$

`Out[2]=` $-A\lceil x \rceil + \lceil x \rceil A$

Now it is easy to find the *Green's operator* G by computing $(1 - P)\,T^{\blacklozenge}$ in its normal form:

`In[3]:=` $\text{Compute}[(1 - L + \lceil x \rceil L - \lceil x \rceil R)(-A\lceil x \rceil + \lceil x \rceil A),$
$\quad \text{by} \to \text{GreenEvaluator},$

`Out[3]=` $-A\lceil x \rceil - \lceil x \rceil B + \lceil x \rceil A \lceil x \rceil + \lceil x \rceil B \lceil x \rceil$

Of course, we could also compute the Green's operator *immediately* (by specifying the given differential operator together with the list of boundary operators):

`In[4]:=` $\text{Compute}[\text{Green}[D^2, \langle L, R \rangle, \text{by} \to \text{GreenEvaluator},$

`Out[4]=` $-A\lceil x \rceil - \lceil x \rceil B + \lceil x \rceil A \lceil x \rceil + \lceil x \rceil B \lceil x \rceil$

Using the translation procedure described after Theorem 3, this corresponds to the *Green's function*

$$g(x,\xi) = \begin{cases} (x-1)\xi & \text{if} \quad 0 \le \xi \le x \le 1, \\ x(\xi-1) & \text{if} \quad 0 \le x \le \xi \le 1. \end{cases}$$

3.2 Damped Oscillations

For a slightly more complicated problem, we take Example 2 in Kralle's book (Krall, 1986, page 109); the differential operator of this BVP has *damped oscillations* as its eigenfunctions; see (Krall, 1986, page 107). Stated in our terminology, the problem reads as follows: Given $f \in C^\infty[0,\pi]$, find $u \in C^\infty[0,\pi]$ such that

$$\begin{aligned} &u'' + 2u' + u = f, \\ &u(0) = u(\pi) = 0 \end{aligned}$$

This time, we will immediately compute the *Green's operator*:

`In[5]:=` Compute[Green[$D^2 + 2D + 1, \langle L, R\rangle$,
by $\to$ GreenEvaluator]

`Out[5]=` $(1-\pi^{-1})\lceil e^{-x}x\rceil A\lceil e^{x}\rceil - \lceil e^{-x}\rceil A\lceil e^{x}x\rceil + \pi^{-1}\lceil e^{-x}x\rceil A\lceil e^{x}x\rceil$
$- \pi^{-1}\lceil e^{-x}x\rceil B\lceil e^{x}\rceil + \pi^{-1}\lceil e^{-x}x\rceil B\lceil e^{x}x\rceil$

Written in the language of *Green's functions*, this means that

$$g(x,\xi) = \begin{cases} \frac{1}{\pi}\,(\pi - x)\,\xi\, e^{\xi-x} & \text{if} \quad 0 \le \xi \le x \le \pi, \\ \frac{1}{\pi}\,(\pi - \xi)\,x\, e^{\xi-x} & \text{if} \quad 0 \le x \le \xi \le \pi. \end{cases}$$

3.3 Transverse Beam Deflection

As a final example, let us do a fourth-order problem (Stakgold, 1979, page 49) describing the *transverse deflection* $u \in C^\infty[0,1]$ of a homogeneous beam with given distributed transversal load $f \in C^\infty[0,1]$, simply supported at both ends. Using a linear elasticity model, one ends up with

$$\begin{aligned} &u'''' = f, \\ &u(0) = u(1) = u''(0) = u''(1) = 0. \end{aligned}$$

Again computing the *Green's operator* directly, we end up with:

$$\texttt{In[6]:=}\quad \mathrm{Compute}[\mathrm{Green}[D^4, \langle L, R, LD^2, RD^2\rangle,\\ \quad\quad \mathrm{by} \rightarrow \mathrm{GreenEvaluator}]$$

$$\begin{aligned}\texttt{Out[6]=}\quad & \tfrac{1}{3}\lceil x\rceil A\lceil x\rceil - \tfrac{1}{6}A\lceil x^3\rceil - \tfrac{1}{2}\lceil x^2\rceil A\lceil x\rceil + \tfrac{1}{6}\lceil x\rceil A\lceil x^3\rceil \\ & +\tfrac{1}{6}\lceil x^3\rceil A\lceil x\rceil + \tfrac{1}{3}\lceil x\rceil B\lceil x\rceil - \tfrac{1}{2}\lceil x\rceil B\lceil x^2\rceil \\ & -\tfrac{1}{6}\lceil x^3\rceil B + \tfrac{1}{6}\lceil x\rceil B\lceil x^3\rceil + \tfrac{1}{6}\lceil x^3\rceil B\lceil x\rceil\end{aligned}$$

This corresponds to the *Green's function*

$$g(x,\xi) = \begin{cases} \frac{1}{3}x\,\xi - \frac{1}{6}\xi^3 - \frac{1}{2}x^2\xi + \frac{1}{6}x\,\xi^3 + \frac{1}{6}x^3\xi & \text{if} \quad 0 \le \xi \le x \le 1, \\ \frac{1}{3}x\,\xi - \frac{1}{2}x\,\xi^2 - \frac{1}{6}x^3 + \frac{1}{6}x\,\xi^3 + \frac{1}{6}x^3\xi & \text{if} \quad 0 \le x \le \xi \le \pi. \end{cases}$$

4 Conclusion

Judging from the applied point of view, what we have presented in this paper is of course not—yet—very exciting. We have only considered a rather *narrow and simple class of BVPs*, namely regular ones for scalar linear ordinary differential equations. However, we believe that there are some prospects for generalizing our approach. Naturally, the work necessary for this will become increasingly more difficult as one climbs up the ladder of generalizations; but we hope this work will be rewarded by a proportional increase of deep mathematical substance.

Let us first look at some straight-forward *generalizations*; we have discussed most of these also in (Rosenkranz, Buchberger, Engl, 2003).

- We can investigate *systems of differential equations* (together with their boundary conditions) instead of a single one. In the linear case, the resulting theory is very similar to scalar BVPs, using a Green's matrix instead of a Green's function; see e.g. page 249 in Kamke (1983). Our method should be extensible to this case in a fairly simple manner. In the worst case, we have to recede to our original approach in (Rosenkranz, Buchberger, Engl, 2003) via Gröbner bases and adapt them to work for vectors of polynomials rather than single ones. Essentially this amounts to computing Gröbner bases in modules, which is a routine task for commutative polynomials—see e.g. (Becker, Weispfenning, 1993, pages 485ff)— and should smoothly carry over to noncommutative ones.
- It is certainly a much greater challenge to move from ordinary to *partial differential equations.* In principle, the algebraization employed in our approach extends in a straight-forward way, e.g. introducing D_x and D_y instead

of the single differentiation D and analogous operators for integration. Here one might be able to benefit a lot from the algebraic approach employed in Riquier-Janet theory and from the symmetry methods of Lie analysis. The treatment of boundary values must of course be adapted. Besides this, the analog of right inversion will be far more complex for most partial differential operators; it might be analogous to the elimination techniques used in the holonomic approach (Zeilberger, 1990).

- One of the most difficult generalizations is probably the step towards *nonlinear* BVPs. The reason is that our algebraic model does not lend itself easily to describe nonlinear differential operators, and a systematic approach might lead to general rewriting (still with respect to the polynomial congruence), where one needs substitution in addition to replacement. Maybe this could be handled by a suitable combination of Gröbner bases and the Knuth-Bendix algorithm; see (Bachmair, Ganzinger, 1994) and (Marche, 1996).
- In this thesis we have only considered regular BVPs in the sense that there is a unique solution, and in this case the Moore-Penrose inverse coincides with the actual inverse. If the BVP is *underdetermined*, however, one can still search for a so-called modified Green's function; see (Stakgold, 1979, page 215). Since the modified Green's function just corresponds to a Moore-Penrose inverse, our method should be adaptable to this case in a natural way.
- As a kind of curiosity, we should also be able to handle certain *integro-differential equations*. In fact, the Green's algebra provides a uniform way of expressing integral as well as differential equations—and their mixtures.

Beyond these rather direct continuations of the research topic treated in this thesis, we believe that our approach has some intrinsic interest not directly tied to BVPs of any kind. The essence of our method can be described as solving problems at the operator level via polynomial methods. This could be a new research paradigm applicable to various problems of a field that might be called *symbolic functional analysis*. Up to now, symbolic methods have conquered the following two "main floors": numbers (computer algebra) and functions (computer analysis); naturally, the third floor would be: operators (symbolic functional analysis). We have described these ideas in more detail in (Buchberger, Engl, 2003); so let us just mention here two examples of problems residing on this third floor:

- Certain problems in *potential theory* have a flavor that is very similar to that of BVPs for PDEs, at least when seen from the symbolic viewpoint. It is therefore natural to ask in how far one could transfer some ideas from BVPs to the potential setting. In particular, one would like to formulate an algebraic setup that allows to express the operator induced by the potential function (analogous to the Green's operator induced by the Green's function).

- The field of *inverse problems* (Engl, Hanke, Neubauer, 1996) opens a whole arena of possible applications for methods of symbolic functional analysis. Even though one cannot usually expect algebraic solutions for such problems, the polynomial approach will certainly uncover a great deal about the solution manifold. In particular, it might be possible to transform the given problem into a different one possessing more profitable properties.

Pondering these examples, we do hope that it will be possible to develop some fruitful ideas along these lines in the near future.

References

L. Bachmair, H. Ganzinger, Buchberger's algorithm: A constraint-based completion procedure. In J.-P. Jouannaud (ed.), *First International Conference on Constraints in Computational Logis*, volume 845 of Lecture Notes in Computer Science, Springer, New York, 1994, pages 285–301.

Thomas Becker, Volker Weispfenning, *Gröbner Bases: A Computational Approach to Commutative Algebra*, Graduate Texts in Mathematics, Springer, New York, 1993.

George M. Bergman, The Diamond Lemma for Ring Theory, *Adv. Math.*, **29**, 1978, pages 179–218.

Bruno Buchberger, *An Algorithm for Finding a Basis for the Residual Class Ring of Zero-Dimensional Polynomial Ideal* (in German), PhD Thesis, University of Innsbruck, Institute for Mathematics, 1965.

Bruno Buchberger, An Algorithmic Criterion for the Solvability of Algebraic Systems of Equations (in German), *Æquationes Mathematicae*, **4**, 1970, pages 374–383.

Bruno Buchberger, *'Introduction to Gröbner Bases*, in Buchberger, Winkler (1998), pages 3–31.

Bruno Buchberger, Franz Winkler (eds.), *Gröbner Bases and Applications*, number 251 in London Mathematical Society Lecture Notes, Cambridge University Press, Cambridge (UK), 1998.

Bruno Buchberger, Heinz W. Engl, emphComputer Algebra for Pure and Applied Functional Analysis. An FWF Proposal for a New SFB Project (F1322?), 2003.

F. Chyzak, B. Salvy, Non-commutative Elimination in Ore Algebras, *Journ. Symb. Comp.* **11**, 1997, pages 187–227.

E. A. Coddington, N. Levinson, *Theory of Ordinary Differential Equations*, McGraw-Hill Book Company, New York, 1955.

H. W. Engl, M. Z. Nashed, New Extremal Characterizations of Generalized Inverses of Linear Operators, *Journal of Mathematical Analysis and Applications*, **82** (1981), pages 566–586.

H.W. Engl, M. Hanke, A. Neubauer, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.

J.W. Helton, Robert L. Miller, *The System NCAlgebra*, homepage at http://math.ucsd.edu/~ncalg.

J.W. Helton, M. Stankus, J. Wavrik, Computer Simplification of Engineering Systems Formulas, *IEEE Trans. Autom. Control*, **43**(3), 1998, pages 302–314.

E. Kamke, *Differentialgleichungen und Lösungsmethoden (Volume 1)*, Teubner, Stuttgart, tenth edition, 1983.

A.M. Krall, *Applied Analysis*, D. Reidel Publishing Company, Dordrecht, 1986.

C. Marché, Normalized Rewriting: An Alternative to Rewriting Modulo a Set of Equations, *Journ. Symb. Comp.*, **11**, 1996, pages 1–36.

Theo Mora, *Gröbner Bases for Non-commutative Polynomial Rings.* In Jacques Calmet (ed.), AAECC-3, Lecture Notes of Computer Science 229, Springer, Berlin, 1986, pages 353–362.

Theo Mora, *Seven Variations on Standard Bases.* Preprint 45, March 1988.

M. Z. Nashed (ed.), *Generalized Inverses and Applications*, Proceedings of an Advanced Seminar Sponsored by the Mathematics Research Center, Academic Press, New York, 1976.

Markus Rosenkranz, Bruno Buchberger, Heinz W. Engl, Solving Linear Boundary Value Problems Via Non-commutative Gröbner Bases. *Applicable Analysis*, **82**(7), July 2003, pages 655–675.

Markus Rosenkranz, *The Green's Algebra: A Polynomial Approach to Boundary Value Problems.* PhD Thesis, June 2003. Johannes Kepler University, Research Institute of Symbolic Computation, A-4040 Linz, Austria.

I. Stakgold, *Green's Functions and Boundary Value Problems*, John Wiley & Sons, New York, 1979.

V. Ufnarovski, *Introduction to Noncommutative Gröbner Bases Theory*, in Buchberger, Winkler (1998), pages 259–280.

D. Zeilberger. A holonomic systems approach to special functions identities, *Journal of Computational and Applied Mathematics*, **32**, 1990, pages 331–368.