Solving Linear Boundary Value Problems via Non–Commutative Gröbner Bases*

Markus Rosenkranz[1] and Heinz W. Engl[2]

[1] *Research Institute for Symbolic Computation, Kepler University, A–4020 Linz, Austria*
[2] *Institute for Industrial Mathematics, Kepler University, A–4020 Linz, Austria*

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

Abstract

A new approach for symbolically solving linear boundary value problems is presented. Rather than using general–purpose tools for obtaining parametrized solutions of the underlying ODE and fitting them against the specified boundary conditions (which may be quite expensive), the problem is interpreted as an operator inversion problem in a suitable Banach space setting. Using the concept of the oblique Moore–Penrose inverse, it is possible to transform the inversion problem into a system of operator equations that can be attacked by virtue of non–commutative Gröbner bases. The resulting operator solution can be represented as an integral operator having the classical Green's function as its kernel. Although, at this stage of research, we cannot yet give an algorithmic formulation of the method and its domain of admissible inputs, we do believe that it has promising perspectives of automation and generalization; some of these perspectives are discussed.

KEYWORDS: Symbolic Methods for ODE, Linear BVP, Moore–Penrose Equations

1. Introduction

Sophus Lie said in 1894 what is nowadays folklore [13, p. 488]: "All branches of physics pos problems that end up in integrating differential equations," and similar things can be said about many other sciences. A great deal of these differential equations come in the form of *boundary value problems*, and it is this problem type that has inspired rich parts of functional analysis, as one can see nicely in the classic work of Hilbert–Courant [9].

It is therefore natural to ask about *symbolic methods* for boundary value problems (BVP). But quite in contrast to the rich arsenal of numerical algorithms for BVP, this corner of mathematics seems to be a bit neglected by the "symbolic world". Of course, there are some standard techniques available for various kinds of differential equations—ordinary and partial, linear and nonlinear [10][23][19]. At the first glance, one might think this is sufficient since one can always solve corresponding differential equation and adapt the free coefficients of the generic solution to fit the boundary conditions. However, we are not only asking for the solution of one individual differential equation generated by fixing the inhomogeneity on its right–hand side; what we want is a generic expression that can be instantiated by all admissible right–hand sides for producing the corresponding solutions (see below). Besides this, the generic solution might have no closed form whereas its "adaption" to the given boundary conditions often does.

Therefore we propose a new approach that works on the BVP as a whole, representing both the differential equation and the boundary conditions by operators on suitable Banach spaces. Such a *functional–analytic setting* is of course very familiar in abstract convergence analysis of numerical BVP algorithms, but interestingly it turns out to be equally useful for searching symbolic solutions via non–commutative Gröbner bases. The idea is that both the differential and the boundary operator are built up from some "atomic" operators and can thus be seen as non–commutative polynomials with the atomic operators as its indeterminates. For obtaining suitable polynomial equations, we use the powerful concept of the oblique Moore–Penrose inverse [21].

In this paper, we consider only *ordinary* differential operators and *linear* BVP; see Section 4 for a discussion of possible extensions. Furthermore we will search for solutions over a *finite interval* $[a, b]$. Now let T be a linear differential operator of order n, so for $u \in C^n[a, b]$ we have

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

where $c_0, \ldots, c_n$ are sufficiently smooth coefficient functions (for example, $c_j \in C^{n-j}[a, b]$ for each $j = 0, \ldots, n$) and c_0 does not vanish. We view T as a linear operator on the Banch space $(C[a, b], \|\cdot\|_\infty)$ with dense domain of definition $\mathcal{D}(T) = C^n[a, b]$. The boundary operators $B_1, \ldots, B_n$ are defined on the same domain; for each $i = 1, \ldots, n$ we have

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

where the coefficients $p_{i,j}, q_{i,j}$ are real numbers. Now the *boundary value problem induced by T and $B_1, \ldots, B_n$* is to find for each right–hand side $f \in C[a, b]$ a function $u \in C^n[a, b]$ such that:

$$\begin{aligned} & T\,u = f \\ & B_1\,u = \ldots = B_n\,u = 0 \end{aligned} \qquad (1)$$

This BVP is actually inhomogeneous in the differential equation and homogeneous in the boundary conditions (*semi–inhomogeneous problem*). But we can always decompose a fully inhomogeneous problem into such a semi–inhomogeneous one and a rather trivial BVP with homogeneous differential equation and inhomogeneous boundary conditions (*semi–homogeneous problem*); see [24, p. 43] for an explanation. Furthermore, we will assume throughout the paper that the boundar conditions are such that they determine a unique solution u of (1) for all $f \in C[a, b]$.

We are now searching for an operator G that takes the inhomogeneity f as input and produces the solution u of (1) as output. In fact, in those cases which we consider, it is well–known that the operator G can be written as an integral operator with the so–called *Green's function g* as its kernel [8, p. 296]:

$$G f(x) = \int_a^b g(x, \xi)\, f(\xi)\, d\xi \tag{2}$$

The desired solution operator G is obviously a *right inverse* of the given differential operator: $T\, G(f) = f$ and hence $T\, G = 1$. (For the sake of simplicity, we will use the symbol 1 for denoting various identity functions and operators.) Of course, there are many right inverses for T, but the boundary conditions $B_1\, u = \ldots = B_n\, u = 0$ are supposed to single out the one we want. It should be noted that this viewpoint is different from the standard one, where the boundary conditions are used for specifying the domain of the differential operator; in this case, there is of course only one inverse.

So we want to find a right inverse that is normally not an inverse in the strict sense—this is where the concept of the *oblique Moore–Penrose inverse* enters the stage (see Section 2 for details): Given the operator T on the Banach space $C[a, b]$ together with arbitrary projectors P, Q onto its nullspace and range closure, the oblique Moore–Penrose inverse $T^{\dagger}_{P,Q}$ can be determined by the four well–known Moore–Penrose equations, which can be seen as four non–commutative polynomial equations in the indeterminates $T, T^{\dagger}, P, Q$. By choosing suitable projectors P, Q, it may be possible to enforce the boundary conditions, which has the consequence that $T^{\dagger} = G$. In general, the projectors will thus become polynomials in $B_1, \ldots, B_n$ and some extra operators describing their particular structure. In many cases, one will be able to express some or all of the boundary operators as well as the differential operator T in terms of these extra operators. So let $A_1, \ldots, A_m$ be those boundary and extra operators that are needed; we will collectively call them *auxiliary operators*. Substituting them in the Moore–Penrose equations, we will end up with an equation system

$$\forall_{i=1,\ldots,4}\ \mathcal{P}_i(G, A_1, \ldots, A_m) = 0\,, \tag{3}$$

where $\mathcal{P}_1, \ldots, \mathcal{P}_4$ are some non–commutative polynomials in the indicated indeterminates.

Our goal is to obtain a partial triangularization this system, i.e. to find an equivalent system containing an equation of the form $G = \ldots$, where the right–hand side should not contain G. This means we want a term representation for the solution operator G: it should be described in terms of some elementary operators like integration and multiplication. For giving a complete specification, we must therefore decide which *elementary operators* $E_1, \ldots, E_k$ we want to allow in the solution term for G. Depending on this choice, the task of triangularizing the equation system may be easy, difficult or even impossible. This is one of the critical points in our approach that should become algorithmic in the future (see Section 4 for a brief discussion of this topic): We must either be creative in finding "good" elementary operators or we need powerful structure theorems for warranting the completeness of certain basis operators.

Assuming we have established a suitable collection of elementary operators $E_1, \ldots, E_k$, we must still specify how they are related with the auxiliary operators $A_1, \ldots, A_m$ occurring in the Moore–Penrose equations, i.e. we need some polynomial equations that describe their interaction. For example, if E_1 is integration and A_1 is differentiation, the obvious relation between them is the Fundamental Theorem of Calculus. This step is the second half of the "creative" phase just described; both steps should be taken together. Having found enough *interaction equations*

$$\forall_{i=1,\ldots,l}\ \mathcal{Q}_i(A_1, \ldots, A_m, E_1, \ldots, E_k) = 0\,, \tag{4}$$

we can combine (3) and (4), looking at it as a well–known problem of computer algebra: Given the ideal J induced by the polynomials $\mathcal{P}_1, \ldots, \mathcal{P}_4, \mathcal{Q}_1, \ldots, \mathcal{Q}_l$, try to find a basis for J containing a polynomial with leading term G; see at the end of Section 2 for an example. Having such a basis, we can write the corresponding equation in the desired form $G = \mathcal{G}(A_1, \ldots, A_m, E_1, \ldots, E_k)$, where $\mathcal{G}$ is a polynomial in the indicated indeterminates. If we have chosen suitable operators $A_1, \ldots, A_m, E_1, \ldots, E_k$, we can interpret the solution operator G as the usual Green's operator and extract from it the Green's function g.

For finding the desired basis, we use the *method of Gröbner bases*, introduced by the second author in his PhD thesis [3]; see also the journal version [4] and a concise treatment in [6]. The advantage of Gröbner bases is that they do not only lead to the desired solution but they also reveal useful information about the ideal structure. In this paper, however, we will not address these issues. For a modern survey of the theory of Gröbner bases and their applications, see [7] and the remarks at the end of this section.

The idea of using the *Moore–Penrose inverse for solving linear BVP* is not new. One can find a standard treatment of this subject in [22] and [24]. But what is new, to our knowledge, is the observ tion that by means of non–commutative Gröbner bases one can actually fertilize the Moore–Penrose equations for obtaining symbolic solutions. There is an interesting paper [18] from the seventies tha describes a different Moore–Penrose method for approaching linear BVP. It is based on the concept of adjoint operators and orthogonal projectors (as opposed to the oblique ones used in our method), but it does not make use of Gröbner bases. This approach seems to result into more complex computations than ours, but it would be an interesting research topic to combine the two methods.

Non–commutative Gröbner bases have been applied to differential operators for several decades, see for example the survey article [24] about Gröbner bases and partial differential equa tions. However, most of the theory in this field is concerned with studying the structure of solutions, without giving explicit methods for constructing them (the situation becomes even worse when it comes to BVP). Besides this, Gröbner bases have been used for *simplifying complicated operator expressions* as they typically arise in control theory. This approach is described in the papers [15][16][26] of the San Diego group, which also served as the starting point for our investigations. used the software package developed by their group for the Gröbner–basis computations necessary in our examples; see Section 2 for details.

The difference between the problem considered here and the subject of simplification addressed by their group is of a fundamental nature. Applications of Gröbner bases—both in the commutative and in the non–commutative cases—come in *three main categories* [6]:

- *Confluent Rewriting*: A Gröbner basis induces a rewrite system for reducing polynomials. Using a suitable term ordering, this will sometimes lead to a drastically simpler optical appearance, which is very important for control theorists [26]. However, the essential poir is that the reduced form is not only optically simpler but even canonical, due to the characteristic Church–Rosser property of Gröbner basis. This means that one can decide equality: Two polynomials are equal in the given ideal if and *only* if their reduced forms are identical.

- *Polynomial Equation Solving*: Using a term ordering of the lexicographic type, Gröbner bases enjoy the so–called elimination property. Basically this means that the equation system will be triangularized as much as possible so that it is easy to solve the resulting system. The elimination property also holds in the non–commutative case; see [2].

- *Syzygies*: The information contained in the reductions that transform a given set of polynomi als into a Gröbner basis can be used to determine the complete solution module of a linear equation system over a polynomial ring.

Seen in this way, research in the San Diego group belongs to the first category whereas our research belongs to the second. It might be worthwhile to also carry out operator–theoretic investigations in fields pertaining to the third application category.

The rest of the paper is *structured as follows*: In Section 2 we take a well–known linear BVP as a simple but yet interesting example for walking through the whole procedure outlined above. In Section 3 we briefly present some more examples demonstrating different boundary conditions and slightly more complicated differential equations. In Section 4 we conclude with some reflections about the methodology and the potential of automation and generalization.

2. A Detailed Computation

The following problem seems to be one of the classical examples that are most often used for introducing the concepts of linear BVP [24, p. 42]. It can be interpreted as describing *one–dimen sional steady heat conduction in a homogeneous rod*. We will discuss this example in some detail for illustrating the solution strategy presented in the previous section.

Given: $f \in C[0, 1]$,

find: $u \in C^2[0, 1]$

such that

$$u'' = f \,,$$
$$u(0) = u(1) = 0 \,.$$

The general problem described above is now given the simple instantiation $[a, b] = [0, 1]$, $n = 2$, $T = D^2$, $B_1 = L$, $B_2 = R$. Here D^2 denotes the iterated differentiation operator on the Banach space $(C[0, 1], \|\cdot\|_\infty)$; it has the subset $C^2[0, 1]$ as its dense domain of definition. The left and right boundary operators L, R are defined in the obvious way: For each $u \in C[0, 1]$, we have $L\,u = u(0)$ and $R\,u = u(1)$. As described above, we interpret this as an *inversion problem* in the following sense: Find a right inverse G of the operator D^2 such that the boundary conditions are also fulfilled. We construct G as a Moore–Penrose inverse.

From the theory [12, p. 567] it is clear that we must fix *appropriate projectors* P and Q onto th nullspace and range–closure of D^2, respectively. The latter will always be the identity $1 : C[0, 1] \to C[0, 1]$ for the type of problems considered here; as a consequence, G is bounded and defined on all of $C[0, 1]$. The other projector P maps $C[0, 1]$ onto the *nullspace*

$$N = \mathcal{N}(D^2) = \{x \mapsto \alpha\, x + \beta \mid \alpha, \beta \in \mathbb{R}\},$$

so choosing P amounts to specifying for each $u \in C[0, 1]$ real numbers α, β such that $(P\,u)(x) = \alpha\, x + \beta$ for all $x \in [0, 1]$. We use this freedom to ensure the boundary conditions, which leads to

$$P = (1 - X)\,L + X\,R\ ,$$

where X is defined as the operator induced by multiplication with the independent variable.

Substituting this together with Q into the general equations in [12, p. 567], we arrive at th following *concrete Moore–Penrose equations*:

$$\begin{aligned} &D^2\,G\,D^2 = D^2 \\ &G\,D^2\,G = G \\ &G\,D^2 = 1 - (1 - X)\,L - X\,R \\ &D^2\,G = 1 \end{aligned} \tag{5}$$

We can see that they form indeed a *system of polynomial equations*, having the desired Green's operator G and the auxiliary operators (named $A_1, \ldots, A_m$ in the introduction) D, X, L, R as indeterminates. The only thing missing now are the elementary operators (named $E_1, \ldots, E_k$ in the introduction) that we want to allow in the solution term, together with suitable relations describing their interaction with the auxiliary operators.

Now we come to the "creative" step of our approach (see Section 4 for a brief discussion on the potential of automation). It is clear that the operators D, X, L, R will not be sufficient for expressing the solution term for the Green's operator G. Since we would like to have an integral representation for G, having the corresponding Green's function g as its kernel, we must obviously take the *antiderivative operator* A as one elementary operator. It is defined in the obvious way as

$$(A\,u)\,x = \int_0^x u(\xi)\, d\xi$$

for all $u \in C[0, 1]$ and $x \in [0, 1]$. What other elementary operators might be needed? In view of the duality in the boundary operators L, R, we may have the idea of adding the operator B *adjoint to the antiderivative operator* A. Whereas the operator A integrates *from* the left boundary, the operator B integrates *to* the right boundary, so it is defined as

$$(B\,u)\,x = \int_x^1 u(\xi)\, d\xi$$

again for all $u \in C[0, 1]$ and $x \in [0, 1]$. Having A and B as elementary operators along with the auxiliary operators D, X, L, R, it turns out that we can express the solution G in the desired way. The following *interaction equations* are sufficient for describing their relations:

$$\begin{aligned} &D\,X = X\,D + 1 \\ &D\,A = 1 \\ &A\,D = 1 - L \\ &D\,B = -1 \end{aligned} \tag{6}$$

$$BD = R - 1$$
$$RX = R$$
$$LX = 0$$

At this point, we have assembled the complete polynomial equation system, consisting of the polynomials $\mathcal{P}_1, \ldots, \mathcal{P}_4$ in the concrete Moore–Penrose equations (5) and the polynomials $\mathcal{Q}_1, \ldots, \mathcal{Q}_7$ of the interaction equations (6). Our goal is to solve this system for G, i.e. we want to find the elimination ideal with respect to G. For this we will use the following *multigraded lexicographic term ordering*:

$$D < R < L < X < A < B \ll G$$

For computing the desired elimination ideal, we use the system *NCAlgebra* [14], a Mathematic package for doing non–commutative computer algebra, written by J. William Helton (Mathematics Department of the University of California, San Diego, California) and Robert L. Miller (General Atomic Corporation, La Jolla, California). It includes support for non–commutative Gröbner bases, also described in the papers [15][16][26]. Typically, we must content ourselves with a partial basis, this is sufficient for us as long as G is isolated. For the current problem *NCAlgebra* returns the following answer:

```
final = NCMakeGB[initial, 2] // ColumnForm
−1 + D ** A
−1 + L + A ** D
1 + D ** B
1 − R + B ** D
1 − D ** X + X ** D
−R + R ** X
L ** X
L ** A
−A − B + R ** A
−A − X + D ** X ** A
…
G + A ** X + X ** B − X ** A ** X − X ** B ** X
−R + L ** R
−R + D ** X ** R
−A − B + B ** A + B ** X + X ** A
…
```

The system has produced 42 polynomials, most of which are left out above as they are not interesting for our present purposes; e.g. some of them express integration rules for polynomials such as $2x^2 + 3x$. The only important thing is that there is only *one polynomial involving the solution operator* G, and in this polynomial, G does indeed occur isolated. Writing the result in the usual format, we arrive at:

$$G = XAX - AX + XBX - XB$$

It is straightforward to rewrite this polynomial into the traditional formulation of the corresponding *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. Other Examples

Passing on to other examples, let us first remark that we can use *various other types of boundary conditions* without making essential changes in the computation just presented. For example, using the mixed conditions $u'(0) = u(1) = 0$ will lead to the nullspace projector

$$P = X\,L\,D + R + L\,D\,,$$

whereas the conditions $u(0) = u'(0) = 0$ will lead to the nullspace projector

$$P = X\,L\,D + L\,.$$

Everything else remains the same, and the computation results in the correct Green's functions for these cases. (Specifying the boundary conditions $u'(0) = u'(1) = 0$, however, would not allow a unique solution for all right–hand sides $f \in C[0, 1]$. In fact, one can apply the well–known *Fredholm alternative* for characterizing solvability in such cases. If we tried to apply our method to such a case, we would end up with redundant parameters. New ideas are necessary for dealing with these cases, but we will not address them here.)

For a slightly more complicated problem, we take Example 2 in Kralle's book [17, p. 109]. Th differential operator of this BVP has *damped oscillations* as its eigenfunctions [17, p. 107]. Stated i our terminology, the problem reads as follows:

$$-\frac{(e^{2x}\,u(x)')' + e^{2x}\,u(x)}{e^{2x}} = f(x)$$
$$u(0) = u(\pi) = 0$$

Here x is assumed to range over the interval $[0, \pi]$. The notation $\boldsymbol{T}'$ is an abbreviation for $\frac{d}{dx}\,\boldsymbol{T}$, where the differential quantifier $\frac{d}{dx}$ operates on the term $\boldsymbol{T}$. For obtaining an operator equation, we introduce some *auxiliary operators*. For all $u \in C[0, \pi]$, $v \in C^1[0, \pi]$ and $x \in [0, \pi]$, we define:

$$(D\,v)(x) = v(x)'$$
$$(E\,u)(x) = e^x\,u(x)$$
$$(F\,u)(x) = e^{-x}\,u(x)$$
$$(L\,u)(x) = u(0)$$
$$(R\,u)(x) = u(\pi)$$

Using these operators, the given BVP can be stated in the following *operator–theoretic form*:

$$-(F^2 D E^2 D + 1) u = f$$
$$L u = R u = 0$$

Going through the procedure explained above, one finds for the nullspace projector

$$P = \frac{e^\pi}{\pi} X F R - \frac{1}{\pi} X F L + F L,$$

It turns out that one does not need other operators except A and B as in the previous examples, but one must add some obvious interaction equations for the new operators E and F. Carrying out the computation in *NCAlgebra*, one obtains the following result (after applying some tedious tricks for representing the "commuting variables" e, π):

$$G = F A X E + X F B E - \frac{1}{\pi} X F A X E - \frac{1}{\pi} X F B X E =$$
$$\frac{1}{\pi} (\pi - X) F A X E + \frac{1}{\pi} X F B (\pi - X) E$$

This time, the partial basis contains 164 polynomials, but there is still only one among which involves G, namely exactly the one corresponding to the solved equation above. Going through the usual translation procedure, one can write G as an integral operator with the following *Green's function* also given in [17, p. 110]:

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

The method presented here is not restricted to the classical setting of second–order Sturm–Liouville theory. For seeing this, we take a practically relevant fourth–order problem [17, p. 49], whic describes the *transverse deflection* $u \in C^2[0, 1]$ of a homogeneous beam with distributed transversal load $f \in C[0, 1]$, simply supported at both ends:

$$u^{(4)} = f$$
$$u(0) = u(1) = u''(0) = u''(1) = 0$$

Its *operator–theoretic formulation* is as follows:

$$D^4 u = f$$
$$L u = R u = L D^2 u = R D^2 u = 0$$

Comparing this BVP to the simple heat–conduction problem considered in the beginning, we observe a strong *similarity*. In fact, the only difference is the order of the differential operator and the additional boundary conditions for u'', so we expect that we can use the same auxiliary and elementary functions.

This expectation is indeed fulfilled. Going through the same procedure as in the heat–conduction example, the boundary conditions lead to the following *nullspace projector*:

$$P = \frac{1}{6}\, X^3\,(R\,D^2 - L\,D^2) + \frac{1}{2}\, X^2\, L\, D^2 + \frac{1}{6}\, X\,(6\,R - 6\,L - 2\,L\,D^2 - R\,D^2) + L$$

Using this operator and the interaction equations from the heat–conduction problem, we obtain a polynomial system that can be solved for G. The partial basis returned by *NCAlgebra* consists of 67 polynomials, and exactly one polynomial among them contains the indeterminate G for the *Green's operator*. Written as an equation, this polynomial is:

$$G = \frac{1}{3}\, X\, A\, X - \frac{1}{6}\, A\, X^3 - \frac{1}{2}\, X^2\, A\, X + \frac{1}{6}\, X\, A\, X^3 + \frac{1}{6}\, X^3\, A\, X + \frac{1}{3}\, X\, B\, X - \frac{1}{2}\, X\, B\, X^2 - \frac{1}{6}\, X^3\, B + \frac{1}{6}\, X\, B\, X^3 + \frac{1}{6}\, X^3\, B\, X$$

As usual, we can immediately translate this expression to the more traditional formulation in terms of a *Green's function* g defined thus:

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

This is in full accordance with [17, p. 71], where the result was obtained by means of the causa fundamental solution.

4. Conclusion

We have presented a new *method for solving linear boundary value problems by symbolic techniques*. It proceeds by transforming the given differential equation and its boundary conditions into a system of polynomial equations that can be solved for the desired Green's operator via non–commutative Gröbner bases. Of course, one must specify those operators and properties that should be used for representing the solution term; using the traditional framework of integral operators, one obtains a solution in terms of the usual Green's function. For several examples, we have exhibited suitable interaction equations that lead to a Green's formulation of the solution. (Incidentally, we have also found other representations of the solution, typically involving multiple integrations. Though outside the framework of the traditional Green's functions, these representations may be of numerical interest due to their smoothness properties.)

Let us now briefly analyze the *current status of algorithmization* in this method. In a typical application, it will proceed through the following steps:

- *Derivation of the concrete Moore–Penrose equations*: The major task in this step is to determine the nullspace projector P since we have seen that the range projector Q is always the identity. Substituting P, Q and the given differential operator T in the generic Moore–Penrose equations and renaming the Moore–Penrose inverse $T^\dagger$ into G, we obtain the concrete Moore–Penrose equations. The polynomial for P will contain various auxiliary operators $A_1, \ldots, A_m$, usually made up from parts of the differential and boundary operators.

- *Compilation of the interaction equations*: After selecting suitable elementary operators $E_1, \ldots, E_k$, we have to find sufficiently many equations describing the relations between the auxiliary operators $A_1, \ldots, A_m$ and the elementary operators $E_1, \ldots, E_k$.
- *Computation of a partial Gröbner basis*: The concrete Moore–Penrose equations are combined with the interaction equations and supplied to a non–commutative Gröbner basis system, using a term ordering that isolates the Green's operator G.
- *Extraction of the Green's function*: The Green's operator G obtained in the previous step is transformed into the corresponding Green's function g.

For the first step and the last two steps, our method can be viewed as an algorithm (relative to the solvability of the homogeneous differential equation). For the second step, some *ad–hoc inventions* are still necessary for each problem instance at hand. In particular, one has to provide suitable interaction equations for specifying the solution structure. Some experience in handling BVP should be sufficient for finding these equalities.

Note that, after having found some basic interaction equations, the question of *how and in which order* these equations should be applied is exhaustively answered by the method of Gröbner bases, due to their Church–Rosser property. In a manual calculation, one has to play around with many possible ways of combining equations, which may or may not lead to success. In this sense, an essential portion of the usual "tricks" occurring in manual calculations is covered by our method; the remaining tricks are assoiated with the interaction equations.

We believe that our method can cover various interesting classes of BVP, which we plan to explore in forthcoming papers (including some of the generalizations discussed below). In an ideal situation—presumably hard to achieve—one might approach a systematic search of elementary operators and interaction equations in a manner similar to the *structure theorems* of Liouville theory, which are used for indefinite integration and ordinary differential equations [10, p. 186]. Fixing certain algebraic input domains (e.g. the elementary transcendental functions) for the coefficients of the differential and boundary operators, one might be able to isolate a suitable "Green's domain" $\mathbb{G}$ such that the Green's function g will always be in $\mathbb{G}$. We think that such an expectation is realistic because it is well–known that one can express g in terms of solutions of certain initial value problems. (Note also that we do not claim that the solution of the BVP itself, namely $G f$ for a given right–hand side f, should have any algebraically simple form.)

Having found a Green's domain $\mathbb{G}$, it is probably not difficult to isolate appropriate *elementary operators* along with their *interaction equations*. Some of these operators might be multiplication operators induced by functions from the input domains and $\mathbb{G}$, similar to F and G in Section 3. It should also be observed that most of the interaction equations considered in this paper would come out quite naturally when the elementary operators are introduced in systematic exploration cycles as described in [5].

Finally, let us propose further lines of future research. The problems considered in this paper have some obvious generalizations. Increasing the number of independent variables leads us to BVP for *partial differential equations* like the well–known Dirichlet problem for $\operatorname{div}(a \operatorname{grad} u) = f$. These equations will typically involve differential operators div, grad, rot, … from vector analysis: We can either regard them as operators in their own right or assemble them from the partial differential operators $\partial_x, \partial_y, \partial_z$. The general methodology presented in this paper should be applicable in both cases; identities like the Divergence Theorem of Gauss will take over the role of the Fundamental Theorem of Calculus.

Passing over to *non–linear problems* is a much bigger challenge. In this case, compound operators may not be expressible as polynomials anymore. For example, take the non–linear operator $Q(u)(x) = u'(x)\,u(x)$. One would like to write this operator in terms of an elementary multiplication operator $M(u, v)(x) = u(x)\,v(x)$ as $Q(u) = M(D(u), u)$. Then we would have the product rule as an interaction equation $D \leftrightarrow M$, namely $D\,M(u, v) = M(D(u), v) + M(u, D(v))$. But this is not a purely operator–theoretic description anymore since we cannot get rid of u and v. This means general rewriting is necessary now: we need substitution in addition to replacement (reduction of polynomials is replacement on equivalence classes). Maybe this could be handled by a combination of Gröbner bases and the Knuth–Bendix algorithm. Actually this is a rather subtle topic, but there are promising results recently [1][20].

Orthogonal to these generalizations, one could also investigate *weak solutions*. In this paper, we have only considered classical solutions, but the results also make sense in a more general Sobolev setting. On the one hand, this simply changes domain and codomain of some operators; this does not harm the polynomial formulation since it abstracts from all topological notions. On the other hand, the solution concept itself must be modified by introducing suitable testing functions v and partial integrations. Logically this means that we have a universal quantifier over v on top of the equations, so we cannot take v as an indeterminate. Again, new ideas are necessary.

Apart from these generalizations, there is another issue that may be worth investigating. We have already observed after Equation (5) that the concept of polynomials is not fully adequate for capturing operator composition since it does not restrict the admissible combinations. This becomes even more apparent when we introduce operators like div and grad. In this case, we would like to distinguish vectors from scalars. For example, the composition $\operatorname{div}\operatorname{grad}$ is admissible whereas $\operatorname{div}\operatorname{div}$ does not make sense. But the question of domain adequacy is not of a purely aesthetic nature: It would prevent a great deal of unnecessary S–polynomials during the search for a Gröbner basis. We would need a notion of *restricted polynomials in* $X_1, \ldots, X_n$ such that each indeterminate X_i has an associated domain, $\operatorname{dom}(X_i)$, and codomain, $\operatorname{cod}(X_i)$, where we can build up monomials $X_i\,X_j\,X_k \cdots$ only if $\operatorname{dom}(X_i) = \operatorname{cod}(X_j)$ and $\operatorname{dom}(X_j) = \operatorname{cod}(X_k)$, etc. Since the structure of restricted polynomials is, by its very intention, not closed under multiplication, it figures as an algebraically rather unwieldy concept. It would be interesting to develop some alternative that combines practical needs and algebraic elegance.

References

[1] L. Bachmair, H. Ganzinger. Buchberger's algorithm: A constraint–based completion procedure. In: J.–P. Jouannaud, editor, First International Conference on Constraints in Computational Logics. Volume 845 of Lecture Notes in Computer Science, Springer Verlag, München, 1994, pp. 285–301.

[2] M. A. Borges, Mijail Borges. Gröbner Bases Property on Elimination Ideal in the Noncommutative Case. In [7], pp. 323–337.

[3] B. Buchberger. An Algorithm for Finding a Basis for the Residual Class Ring of Zero–Dimensional Polynomial Ideal (German). PhD Thesis, University of Innsbruck, Institute for Mathematics, 1965.

[4] B. Buchberger. An Algorithmic Criterion for the Solvability of Algebraic Systems of Equations (German). *Aequationes Mathematicae* 4, 1970, pp. 374–383.

[5] B. Buchberger. Theory Exploration Versus Theorem Proving. Invited talk at the Calculemus '99 Conference, Trento, Italy, July 1999. Available as RISC technical report at ftp://ftp.risc.uni–linz.ac.at/pub/techreports/1999/99–46.tar.gz .

[6] B. Buchberger. Introduction to Gröbner Bases. In [7], pp. 3–31.

[7] B. Buchberger, Franz Winkler (eds.). Gröbner Bases and Applications. London Mathematical Society, Lecture Note Series 251, Cambridge University Press 1998.

[8] E. A. Coddington, N. Levinson. Theory of Ordinary Differential Equations. McGraw–Hill Book Company, New York, 1955.

[9] R. Courant, D. Hilbert. Die Methoden der mathematischen Physik, Volumes 1 / 2. Springer Verlag, 4th edition, 1993.

[10] J.H. Davenport, Y. Siret, E. Tournier. Computer Algebra. Academic Press, London 1988.

[11] H.W. Engl, M. Hanke, and A. Neubauer, Regularization of Inverse Problems, Kluwer, Dordrecht, 1996.

[12] H. W. Engl, M. Z. Nashed. New Extremal Characterizations of Generalized Inverses of Linear Operators *Journal of Mathematical Analysis and Applications* **82**, 1981, pp. 566–586.

[13] G. Grosche, V. Ziegler, D. Ziegler. Teubner–Taschenbuch der Mathematik I. Teubner, Stuttgart, 1996.

[14] J. W. Helton, Robert L. Miller. The system NCAlgebra. Homepage at http://math.ucsd.edu/~ncalg, manual at http://math.ucsd.edu/~ncalg/NCBIGDOC/NCBIGDOC.html.

[15] J. W. Helton, Mark Stankus, John Wavrik. Computer Simplification of Engineering Systems Formulas IEEE Trans. Autom. Control 43, No. 3, 302-314 (1998).

[16] J. W. Helton, John Wavrik. Rules for Computer Simplification of the Formulas in Operator Model Theor and Linear Systems. *Operator Theory: Advances and Applications* **73** , 1994, 325—354.

[17] A. M. Krall. Applied Analysis. D. Reidel Publishing Company, Dordrecht, 1986.

[18] W. Loud. Some Examples of Generalized Green's Functions and Generalized Green's Matrices. *SIAM Review*, **12**(2), 1970.

[19] Y.–K. Man. Computing Closed Form Solutions of First Order ODEs Using the Prelle–Singer Procedure *Journ. Symb. Comp.* **16**, 1983, pp. 423–443.

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

[21] M. Z. Nashed. Generalized Inverses and Applications, Proceedings of an Advanced Seminar Sponsored b the Mathematics Research Center, Universiy of Wisconsin–Madison, October 1973. Academic Press, New York 1976.

[22] M. Z. Nashed. Aspects of Generalized Inverses in Analysis and Regularization. In [21], pp. 193–244.

[23] M. J. Prelle, M. F. Singer. Elementary First Integrals of Differential Equations. *Trans. AMS*, **279**(1), 198 pp. 215–229.

[24] D. C. Struppa. Gröbner Bases in Partial Differential Equations. In [7], pp. 235–245.

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

[26] J. Wavrik. Rewrite Rules and Simplification of Matrix Expressions. *Computer Science Journal of Moldova*, **4** (2/11), 1996.