A Symbolic Framework for Operations on Linear Boundary Problems

Markus Rosenkranz[1], Georg Regensburger[1],
Loredana Tec[2], and Bruno Buchberger[2]

[1] Johann Radon Institute for Computational and Applied Mathematics,
Austrian Academy of Sciences, Altenberger Str. 69, 4040 Linz, Austria

[2] Research Institute for Symbolic Computation,
Johannes Kepler Universität, 4032 Castle of Hagenberg, Austria

Abstract. We describe a symbolic framework for treating linear boundary problems with a generic implementation in the Theorema system. For ordinary differential equations, the operations implemented include computing Green's operators, composing boundary problems and integro-differential operators, and factoring boundary problems. Based on our factorization approach, we also present some first steps for symbolically computing Green's operators of simple boundary problems for partial differential equations with constant coefficients. After summarizing the theoretical background on abstract boundary problems, we outline an algebraic structure for partial integro-differential operators. Finally, we describe the implementation in Theorema, which relies on functors for building up the computational domains, and we illustrate it with some sample computations including the unbounded wave equation.

Key words: Linear boundary problem, Green's operator, Integro-Differential Operator, Ordinary Differential Equation, Wave Equation

1 Introduction

Due to their obvious importance in applications, *boundary problems* play a dominant role in Scientific Computing, but almost exclusively in the numerical segment. It is therefore surprising that they have as yet gained little attention in Symbolic Computation, neither from a theoretical perspective nor in computer algebra systems.

In applications [1, p. 42] one is "concerned not only with solving [the boundary problem] for specific data but also with finding a suitable form for the solution that will exhibit its dependence on the data." In our work, we focus on linear boundary problems (and will henceforth suppress the attribute "linear"). For us, a boundary problem is thus a differential equation with a symbolic right-hand side, supplemented by suitable boundary conditions. Solving it means to determine its *Green's operator*, namely the integral operator that maps the right-hand side to the solution. For a symbolic approach to boundary problems, one has to develop a constructive algebraic theory of integral operators and an algorithmic framework for manipulating boundary conditions.

Such a development was initiated in [2], leading to a symbolic method for computing Green's operators of regular two-point boundary problems with constant coefficients [3]. We extended these results to a *differential algebra setting* in [4], where we also developed a *factorization method* applicable to boundary problems for ordinary differential equations (ODEs). A more *abstract view* on boundary problems and a general factorization theory is described in [5], including in particular partial differential equations (PDEs).

In this paper, we describe a generic *implementation in Theorema* [6] of various operations on boundary problems and integro-differential operators for ODEs (Section 5), exemplified in (Appendix A): computing Green's operators, composing boundary problems and integro-differential operators, and factoring boundary problems. The computations are realized by a suitable noncommutative Gröbner basis that reflects the essential interactions between certain basic operators. Gröbner bases were introduced by Buchberger in [7]. For an introduction to the theory, we refer to [8], for its noncommutative extension to [9].

Moreover, for *PDEs* we present some first steps for making the abstract setting of [5] algorithmic. We develop an algebraic language for encoding the integro-differential operators appearing as Green's operators of some simple two-dimensional boundary problems for PDEs with constant coefficients (Section 4). Using our generic factorization approach, this allows to find the Green's operator of higher-order boundary problems by composing those of its lower-order factors. This idea is exemplified for the unbounded wave equation with a sample computation (Appendix A).

We summarize the necessary *theoretical background* on abstract boundary problems, omitting all technical details and illustrating it for the case of ODEs (Section 2). After explaining the composition and factorization of boundary problems (Section 3), we outline the algebraic structures used for encoding ordinary as well as partial integro-differential operators (Section 4).

For motivating our algebraic setting of boundary problems, we consider first the *simplest two-point boundary problem*. Writing $\mathcal{F}$ for the real or complex vector space $C^\infty[0,1]$, it reads as follows: Given $f \in \mathcal{F}$, find $u \in \mathcal{F}$ such that

$$\boxed{\begin{array}{l} u'' = f, \\ u(0) = u(1) = 0. \end{array}} \tag{1}$$

Let $D\colon \mathcal{F} \to \mathcal{F}$ denote the usual derivation and L, R the two linear functionals $L\colon f \mapsto f(0)$ and $R\colon f \mapsto f(1)$. Note that u is annihilated by any linear combination of these functionals so that problem (1) can be described by $(D^2, [L, R])$, where $[L, R]$ is the subspace generated by L, R in the dual space $\mathcal{F}^*$.

As a second example, consider the following boundary problem for the *wave equation* on the domain $\Omega = \mathbb{R} \times \mathbb{R}_{\geq 0}$, now writing $\mathcal{F}$ for $C^\infty(\Omega)$: Given $f \in \mathcal{F}$, find $u \in \mathcal{F}$ such that

$$\boxed{\begin{array}{l} u_{tt} - u_{xx} = f, \\ u(x,0) = u_t(x,0) = 0. \end{array}} \tag{2}$$

Note that we use the terms "boundary condition/problem" in the general sense of linear conditions. The boundary conditions in (2) can be expressed by the infinite family of linear functionals $\beta_x\colon u \mapsto u(x,0)$, $\gamma_x\colon u \mapsto u_t(x,0)$ with x ranging over $\mathbb{R}$. So we can represent the boundary problem again by a pair consisting of the differential operator $D_t^2 - D_x^2$ and the (now infinite dimensional) subspace generated by β_x and γ_x in $\mathcal{F}^*$.

For ensuring a unique representation of boundary conditions, we take the *orthogonal closure* of this subspace, which we denote by $[\beta_x, \gamma_x]_{x\in\mathbb{R}}$. This is the space of all linear functionals vanishing on the functions annihilated by β_x, γ_x. Every finite dimensional subspace is orthogonally closed, but here, for example, the functionals $u \mapsto \int_0^x u(\eta, 0)\, d\eta$ and $u \mapsto u_x(x,0)$ for arbitrary $x \in \mathbb{R}$ are in the orthogonal closure but not in the space generated by β_x and γ_x. We refer to [10] or [5, App. A.1] for details on the orthogonal closure.

Some *notational conventions.* We use the symbol $\leq$ for algebraic substructures. If $T\colon \mathcal{F} \to \mathcal{G}$ is a linear map and $\mathcal{B} \leq \mathcal{G}^*$, we write $\mathcal{B} \cdot T$ for the subspace $\{\beta \circ T \mid \beta \in \mathcal{B}\} \leq \mathcal{F}^*$. For a subset $\mathcal{B} \subseteq \mathcal{F}^*$ the so-called *orthogonal* is defined as $\mathcal{B}^\perp = \{u \in \mathcal{F} \mid \beta(u) = 0 \text{ for all } \beta \in \mathcal{B}\}$.

2 An Algebraic Formulation of Boundary Problems

In this section, we give a summary of the algebraic setting for boundary problems exposed in [5], see also there for further details and proofs. We illustrate the definitions and statements for ODEs on a compact interval $[a,b] \subseteq \mathbb{R}$. In this setting, most of the statements can be made algorithmic relative to solving homogeneous linear differential equations (and the operations of integration and differentiation).

A *boundary problem* is given by a pair $(T, \mathcal{B})$, where $T\colon \mathcal{F} \to \mathcal{G}$ is a surjective linear map between vector spaces $\mathcal{F}, \mathcal{G}$ and $\mathcal{B} \leq \mathcal{F}^*$ is an orthogonally closed subspace of boundary conditions. We say that $u \in \mathcal{F}$ is a solution of $(T, \mathcal{B})$ for a given $f \in \mathcal{G}$ if $Tu = f$ and $u \in \mathcal{B}^\perp$.

In the *ODE setting*, $T = D^n + c_{n-1}D^{n-1} + \cdots + c_1 D + c_0$ is a monic differential operator of order n with coefficients $c_i \in \mathcal{G}$. For the spaces $\mathcal{F}, \mathcal{G}$ we could for example choose $\mathcal{F} = \mathcal{G} = C^\infty[a,b]$ or $\mathcal{F} = C^n[a,b]$ and $\mathcal{G} = C[a,b]$, as real or complex vector spaces. The differential operator T is surjective since every inhomogeneous linear differential equation has a solution in $\mathcal{F}$, e.g. given by the formula (3) below. The solution space of the homogeneous equation, $\operatorname{Ker} T$, has dimension n, so we require $\dim \mathcal{B} = n$, and we assume that $\mathcal{B}$ is given by a basis $\beta_1, \ldots, \beta_n$. Then the boundary problem reads as follows: Given $f \in \mathcal{G}$, find $u \in \mathcal{F}$ such that

$$\boxed{\begin{array}{l} Tu = f, \\ \beta_1(u) = \cdots = \beta_n(u) = 0. \end{array}}$$

The boundary conditions can in principle be any linear functionals. In particular, they can be point evaluations of derivatives or also more general boundary conditions of the form $\beta(u) = \sum_{i=0}^{n-1} a_i\, u^{(i)}(a) + b_i\, u^{(i)}(b) + \int_a^b v(\xi)\, u(\xi)\, d\xi$ with

$v \in \mathcal{F}$, known in the literature [11] as "Stieltjes boundary conditions". Integral boundary conditions also appear naturally when we factor a boundary problem along a given factorization of the differential operator (Section 3), and they appear in the normal forms of integro-differential operators (Section 4).

A boundary problem $(T, \mathcal{B})$ is *regular* if for each $f \in \mathcal{G}$ there exists exactly one solution u of $(T, \mathcal{B})$. Then we call the linear operator $G\colon \mathcal{G} \to \mathcal{F}$ that maps a right-hand side f to its unique solution $u = Gf$ the *Green's operator* for the boundary problem $(T, \mathcal{B})$, and we say that G solves the boundary problem $(T, \mathcal{B})$. Since $TGf = f$, we see that the Green's operator for a regular boundary problem $(T, \mathcal{B})$ is a right inverse of T, determined by the property $\operatorname{Im} G = \mathcal{B}^\perp$. Therefore we use the notation $G = (T, \mathcal{B})^{-1}$ for the Green's operator.

Regular boundary problems can be characterized as follows. A boundary problem is regular iff $\mathcal{B}^\perp$ is a complement of $\operatorname{Ker} T$ so that $\mathcal{F} = \operatorname{Ker} T \dotplus \mathcal{B}^\perp$ as a direct sum. For ODEs we have the following algorithmic regularity test (compare [12, p. 184] for the special case of two-point boundary conditions): A boundary problem $(T, \mathcal{B})$ for an ODE is regular iff the *evaluation matrix* $B = (\beta_i(u_j))$ is regular, where the β_i and u_j are any basis of respectively $\mathcal{B}$ and $\operatorname{Ker} T$.

Given any right inverse $\tilde{G}$ of a surjective linear map $T\colon \mathcal{F} \to \mathcal{G}$, the Green's operator for a regular boundary problem $(T, \mathcal{B})$ is given by $G = (1 - P)\tilde{G}$, where P is the projector with $\operatorname{Im} P = \operatorname{Ker} T$ and $\operatorname{Ker} P = \mathcal{B}^\perp$. Using this observation, we outline in the following how the Green's operator can be computed in the ODE setting.

Let $(T, \mathcal{B})$ be a regular boundary problem for an ODE of order n with $\mathcal{B} = [\beta_1, \ldots, \beta_n]$, and let $u_1, \ldots, u_n$ be a fundamental system of solutions. We first compute a right inverse of the differential operator T. This can be done by the usual variation-of-constants formula (see for example [13, p. 87] for continuous functions or [14] in a suitable integro-differential algebra setting): Let $W = W(u_1, \ldots, u_n)$ be the Wronskian matrix and $d = \det W$. Moreover, let $d_i = \det W_i$, where W_i is the matrix obtained from W by replacing the ith column by the nth unit vector. Then the solution of the initial value problem $Tu = f$, $u(a) = u'(a) = \cdots = u^{(n-1)}(a) = 0$ is given by

$$u(x) = \sum_{i=1}^{n} u_i(x) {\textstyle\int_a^x} d_i(\xi)/d(\xi)\, f(\xi)\, d\xi. \tag{3}$$

The integral operator $T^\blacklozenge\colon f \mapsto u$ defined by (3) is a right inverse of T, which we also call the *fundamental right inverse*. Computing the projector $P\colon \mathcal{F} \to \mathcal{F}$ with $\operatorname{Im} P = [u_1, \ldots, u_n]$ and $\operatorname{Ker} P = [\beta_1, \ldots, \beta_n]^\perp$ is a linear algebra problem, see for example [5, App. A.1]: Let B be the evaluation matrix $B = (\beta_i(u_j))$. Since $(T, \mathcal{B})$ is regular, B is invertible. Set $(\tilde{\beta}_1, \ldots, \tilde{\beta}_n)^t = B^{-1}(\beta_1, \ldots, \beta_n)^t$. Then the projector P is given by $u \mapsto \sum_{i=1}^n \tilde{\beta}_i(u)\, u_i$. Finally, we compute

$$G = (1 - P)T^\blacklozenge \tag{4}$$

to obtain the Green's operator for $(T, \mathcal{B})$.

3 Composing and Factoring Boundary Problems

In this section we discuss the composition of boundary problems corresponding to their Green's operators. We also describe how factorizations of a boundary problem along a given factorization of the defining operator can be characterized and constructed. We refer again to [5] for further details. In the following, we assume that all operators are defined on suitable spaces such that the composition is well-defined.

Definition 1. *We define the* composition *of boundary problems* $(T_1, \mathcal{B}_1)$ *and* $(T_2, \mathcal{B}_2)$ *by* $(T_1, \mathcal{B}_1) \circ (T_2, \mathcal{B}_2) = (T_1 T_2, \mathcal{B}_1 \cdot T_2 + \mathcal{B}_2)$.

So the boundary conditions from the first boundary problem are "translated" by the operator from the second problem. The composition of boundary problems is associative but in general not commutative. The next proposition tells us that the composition of boundary problems preserves regularity.

Proposition 1. *Let* $(T_1, \mathcal{B}_1)$ *and* $(T_2, \mathcal{B}_2)$ *be regular boundary problems with Green's operators* G_1 *and* G_2*. Then* $(T_1, \mathcal{B}_1) \circ (T_2, \mathcal{B}_2) = (T, \mathcal{B})$ *is regular with Green's operator* $G_2 G_1$ *so that* $((T_1, \mathcal{B}_1) \circ (T_2, \mathcal{B}_2))^{-1} = (T_2, \mathcal{B}_2)^{-1} \circ (T_1, \mathcal{B}_1)^{-1}$.

The simplest example of composing two boundary (more specifically, initial value) problems for ODEs is the following. Using the notation from the Introduction, one sees that $(D, [L]) \circ (D, [L]) = (D^2, [LD] + [L]) = (D^2, [L, LD])$.

Next we write the wave equation (2) as $\mathcal{P} = (D_t^2 - D_x^2, [u(x,0), u_t(x,0)])$, where $u(x,0)$ and $u_t(x,0)$ are short for the functionals $u \mapsto u(x,0)$ and $u \mapsto u_t(x,0)$, respectively, with x ranging over $\mathbb{R}$, and $[\ldots]$ denotes the orthogonal closure of the subspace generated by these functionals. For boundary problems with PDEs, we usually have to describe the boundary conditions as the orthogonal closure of some subspaces that we can describe in finite terms. As detailed in [5], we can still compute the composition of two such problems since taking the orthogonal closure commutes with the operations needed for computing the boundary conditions for the composite problem (precomposition with a linear operator and sum of subspaces).

Using this observation, we can compute $\mathcal{P}$ as the composition of the two boundary problems $\mathcal{P}_1 = (D_t - D_x, [u(x,0)])$ and $\mathcal{P}_2 = (D_t + D_x, [u(x,0)])$ as follows. By Definition 1, we see that $\mathcal{P}_1 \circ \mathcal{P}_2$ equals

$$(D_t^2 - D_x^2, [u_t(x,0) + u_x(x,0)] + [u(x,0)]) = (D_t^2 - D_x^2, [u(x,0), u_t(x,0)]), \quad (5)$$

where the last equality holds since $u(x,0) = 0$ for $x \in \mathbb{R}$ implies also $u_x(x,0) = 0$ for $x \in \mathbb{R}$, showing that $u_x(x,0)$ is in the orthogonal closure $[u(x,0)]$.

In the following, we assume that for a boundary problem $(T, \mathcal{B})$ we have a factorization $T = T_1 T_2$ of the defining operator with surjective linear maps T_1, T_2. In [5], we characterize and construct all factorizations $(T, \mathcal{B}) = (T_1, \mathcal{B}_1) \circ (T_2, \mathcal{B}_2)$ into boundary problems along the given factorization of T. We show in particular that if we factor a regular problem into regular problems, the left factor $(T_1, \mathcal{B}_1)$ is unique, and we can choose for the right factor $(T_2, \mathcal{B}_2)$ any subspace

$\mathcal{B}_2 \le \mathcal{B}$ that makes the problem regular. Moreover, if G_2 is the Green's operator for some regular right factor $(T_2, \mathcal{B}_2)$, the boundary conditions for the left factor can be computed by $\mathcal{B}_1 = \mathcal{B} \cdot G_2$. Factoring boundary problems for differential equations allows us to split a problem of higher order into subproblems of lower order, provided we can factor the differential operator. For the latter, we can exploit algorithms and results about factoring ordinary [15–17] and partial differential operators [18, 19].

For ODEs we can factor boundary problems algorithmically as described in [5] and in an integro-differential algebra setting in [4]. There we assume that we are given a fundamental system of the differential operator T and a right inverse of T_2. As we will detail in the next paragraph, we can also compute boundary conditions $\mathcal{B}_2 \le \mathcal{B}$ such that $(T_2, \mathcal{B}_2)$ is a regular right factor, given only a fundamental system of T_2. We can then compute the left factor as explained above. This can be useful in applications, because it still allows us to factor a boundary problem if we can factor the differential operator and compute a fundamental system of only one factor. The remaining lower order problem can then be solved by numerical methods (and we expect that the integral conditions $\mathcal{B}_1 = \mathcal{B} \cdot G_2$ may be beneficial since they are stable).

Let now $(T, \mathcal{B})$ be a boundary problem of order $m+n$ with boundary conditions $[\beta_1, \ldots, \beta_{m+n}]$. Let $T = T_1 T_2$ be a factorization into factors of respective orders n and m, and let $u_1, \ldots, u_m$ be a fundamental system for T_2. We compute the "partial" $(m+n) \times m$ evaluation matrix $\tilde{B} = \beta_i(u_j)$. Since $(T, \mathcal{B})$ is regular, the full evaluation matrix is regular and hence the columns of $\tilde{B}$ are linearly independent. Therefore computing the reduced row echelon form yields a regular matrix C such that $C\tilde{B} = \binom{I_m}{0}$, where I_m is the $m \times m$ identity matrix. Let now $(\tilde{\beta}_1, \ldots, \tilde{\beta}_{m+n})^t = C(\beta_1, \ldots, \beta_{m+n})^t$ and $\mathcal{B}_2 = [\tilde{\beta}_1, \ldots, \tilde{\beta}_m]$. Then $(T_2, \mathcal{B}_2)$ is a regular right factor since its evaluation matrix is I_m by our construction. See Appendix A for an example.

As a first example, we factor the two-point boundary problem $(D^2, [L, R])$ from the Introduction into two regular problems along the trivial factorization with $T_1 = T_2 = D$. The indefinite integral $A = \int_0^x$ is the Green's operator for the regular right factor $(D, [L])$. The boundary conditions for the unique left factor are $[LA, RA] = [0, RA] = [RA]$, where $RA = \int_0^1$ is the definite integral. So we obtain $(D, [RA]) \circ (D, [L]) = (D^2, [L, R])$ or in traditional notation

$$\boxed{\begin{array}{l} u' = f \\ \int_0^1 u(\xi)\, d\xi = 0 \end{array}} \circ \boxed{\begin{array}{l} u' = f \\ u(0) = 0 \end{array}} = \boxed{\begin{array}{l} u'' = f \\ u(0) = u(1) = 0 \end{array}}.$$

Note that the boundary condition for the left factor is an integral (Stieltjes) boundary condition.

As an example of a boundary problem for a PDE, we factor the wave equation (2) along the factorization $D_t^2 - D_x^2 = (D_t - D_x)(D_t + D_x)$. In Appendix A, we show that one can use this factorization to determine algorithmically its Green's operator. The boundary problem $\mathcal{P}_2 = (D_t + D_x, [u(x, 0)])$ is a regular right factor. In general, choosing boundary conditions in such a way that they make up a regular boundary problem for a given first-order right factor of a linear PDE amounts to a geometric problem involving the characteristics; compare also Section 4. The Green's operator for $\mathcal{P}_2$ is $G_2 f(x,t) = \int_{x-t}^{x} f(\xi, \xi - x + t)\, d\xi$. We can compute the boundary conditions for the left factor by $[u(x,0)\cdot G_2, u_t(x,0)\cdot G_2] = [0, u(x,0)] = [u(x,0)]$ so that $\mathcal{P}_1 = (D_t - D_x, [u(x,0)])$ is the desired left factor. In (5) we have already verified that $\mathcal{P}_1 \circ \mathcal{P}_2 = \mathcal{P}$.

4 Representation of Integro-Differential Operators

For representing ordinary boundary problems as well as their Green's operators in a single algebraic structure, we have introduced the algebra of *integro-differential operators* $\mathcal{F}[\partial, \int]$ in [4], see also [14] for a summary. It is based on integro-differential algebras, which bring together the usual derivation structure with a suitable notion of indefinite integration and evaluation. The integro-differential operators are defined as a quotient of the free algebra in the corresponding operators (derivation, integration, evaluation, and multiplication) modulo an infinite parametrized Gröbner basis. See Section 5 for more details and an implementation. Alternatively, integro-differential operators can also be defined directly in terms of normal forms [20].

Let us now turn to the treatment of *partial differential equations*. We are currently forging an adequate notion of integro-differential operators for describing the Green's operators of an interesting class of PDEs, just as $\mathcal{F}[\partial, \int]$ can be used for ODEs. In the remainder of this section we can only give a flavor (and a small test implementation) of how integro-differential operators for PDEs might look like in a simple case that includes the unbounded wave equation (2).

We construct a ring $\mathcal{R}$ of integro-differential operators acting on the function space $\mathcal{F} = C^\infty(\mathbb{R} \times \mathbb{R})$; for simplicity we neglect here the restriction to $\mathbb{R} \times \mathbb{R}_{\geq 0}$. The ring $\mathcal{R}$ is defined as the free $\mathbb{C}$-algebra in the following indeterminates given with their respective action on a function $f(x,t) \in \mathcal{F}$.

Name	Indeterminates	Action
Differential operators	$D_x,\ D_t$	$f_x(x,t),\ f_t(x,t)$
Integral operators	$A_x,\ A_t$	$\int_0^x f(\xi,t)\, d\xi,\ \int_0^t f(x,\tau)\, d\tau$
Evaluation operators	$L_x,\ L_t$	$f(0,t),\ f(x,0)$
Substitution operators	$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}(\mathbb{R},2)$	$f(ax+bt, cx+dt)$

Similar to the identities governing $\mathcal{F}[\partial, \int]$, described in [4], various relations among the above operators can now be encoded in a quotient of $\mathcal{R}$. We will only sketch the most important relations, focusing on those that are needed for the sample computations. (In a more complete setup, the indeterminates should also be chosen in a more economical way. For example, it is possible to subsume the evaluations under the substitutions if one allows all affine transformations by adding translations and singular matrices.)

First of all, we can transfer all relations from $\mathcal{F}[\partial, \textstyle\int]$ that involve D, A and L, once for the corresponding x-operators and once for the corresponding t-operators. Furthermore, each x-operator commutes with each t-operator. For example, we have $D_x A_x = 1$ but $D_x A_t = A_t D_x$. For normalizing such commutative products, we write the x-operators left of the t-operators. Our strategy for normal forms is thus similar to the case of $\mathcal{F}[\partial, \textstyle\int]$, the only new ingredient being the substitutions: We will move them to the left as much as possible.

Since substitutions operate on the arguments, it is clear that we must reverse their order when multiplying them as elements of $\mathcal{R}$. But the most important relations are those that connect the substitutions with the integro-differential indeterminates: The chain rule governs the interaction with differentiation, the substitution rule with integration. While the former gives rise to the identities

$$D_x M = a\,M D_x + c\,M D_t \quad \text{and} \quad D_t M = b\,M D_x + d\,M D_t$$

for a matrix $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, the relation between M and integrals is a bit subtler. If M is an upper triangular matrix (so that $c = 0$ and $a \neq 0$), the substitution rule yields

$$A_x M = \tfrac{1}{a}(1 - L_x) M A_x,$$

and if M is a lower triangular matrix (so that $b = 0$ and $d \neq 0$) similarly $A_t M = \tfrac{1}{d}(1 - L_t) M A_t$.

But there are no such identities for pushing $\left(\begin{smallmatrix} 1 & 0 \\ c & 1 \end{smallmatrix}\right)$ left of A_x or $\left(\begin{smallmatrix} 1 & b \\ 0 & 1 \end{smallmatrix}\right)$ left of A_t; we leave them in their place for the normal forms. For treating the general case, we make use of a variant of the Bruhat decomposition [21, p. 349], writing $M \in \mathrm{GL}(\mathbb{R}, 2)$ as

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \begin{cases} \left(\begin{smallmatrix} 1 & 0 \\ c/a & 1 \end{smallmatrix}\right) \left(\begin{smallmatrix} a & b \\ 0 & (ad-bc)/a \end{smallmatrix}\right) & \text{if } a \neq 0, \\ \left(\begin{smallmatrix} b & 0 \\ d & c \end{smallmatrix}\right) \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right) & \text{if } a = 0, \end{cases}$$

or alternatively as

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \begin{cases} \left(\begin{smallmatrix} 1 & b/d \\ 0 & 1 \end{smallmatrix}\right) \left(\begin{smallmatrix} (ad-bc)/d & 0 \\ c & d \end{smallmatrix}\right) & \text{if } d \neq 0, \\ \left(\begin{smallmatrix} b & a \\ 0 & c \end{smallmatrix}\right) \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right) & \text{if } d = 0. \end{cases}$$

The former decomposition is applied in deriving the rule for A_x, which reads

$$A_x \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \tfrac{1}{a}\,(1 - L_x) \left(\begin{smallmatrix} a & b \\ 0 & (ad-bc)/a \end{smallmatrix}\right) A_x \left(\begin{smallmatrix} 1 & 0 \\ c/a & 1 \end{smallmatrix}\right)$$

if $a \neq 0$ and otherwise $A_x \left(\begin{smallmatrix} 0 & b \\ c & d \end{smallmatrix}\right) = \tfrac{1}{c}\,(1 - L_x) \left(\begin{smallmatrix} 0 & b \\ c & d \end{smallmatrix}\right) A_t$. Analogously, the latter decomposition yields the rule for A_t as

$$A_t \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \tfrac{1}{d}\,(1 - L_t) \left(\begin{smallmatrix} (ad-bc)/d & 0 \\ c & d \end{smallmatrix}\right) A_t \left(\begin{smallmatrix} 1 & b/d \\ 0 & 1 \end{smallmatrix}\right)$$

if $d \neq 0$ and otherwise $A_t \left(\begin{smallmatrix} a & b \\ c & 0 \end{smallmatrix}\right) = \tfrac{1}{b}\,(1 - L_t) \left(\begin{smallmatrix} a & b \\ c & 0 \end{smallmatrix}\right) A_t$.

According to the rules above, an $\mathcal{R}$-operator like $A_x \left(\begin{smallmatrix} 1 & 0 \\ k & 1 \end{smallmatrix}\right)$ is in normal form. Also $A_x \left(\begin{smallmatrix} 1 & 0 \\ k & 1 \end{smallmatrix}\right) A_x$ is a normal form, describing an area integral. For interpreting it geometrically, it is convenient to postmultiply it with the reverse shear, obtaining thus the integral operator $T_k = \left(\begin{smallmatrix} 1 & 0 \\ -k & 1 \end{smallmatrix}\right) A_x \left(\begin{smallmatrix} 1 & 0 \\ k & 1 \end{smallmatrix}\right) A_x$. One can easily verify that $T_k f(x,t)$ represents the integral of f taken over the triangle with vertices (x,t), $(0,y)$ and $(0, t-kx)$. This is the triangle delimited by the y-axis, the horizontal through (x,y), and the slanted line through (x,t) with slope k. Similar interpretations can be given for products involving A_t.

Finally, we need some rules relating substitutions with evaluations. Here the situation is analogous to the integrals: We can move "most" of the substitutions to the left of an evaluation, but certain shears remain on the right. In detail, we have the rules

$$L_x \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \left(\begin{smallmatrix} 1 & 0 \\ 0 & d \end{smallmatrix}\right) L_x \left(\begin{smallmatrix} 1 & b/d \\ 0 & 1 \end{smallmatrix}\right) \quad \text{if } d \neq 0 \qquad L_x \left(\begin{smallmatrix} a & b \\ c & 0 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 0 & b \\ 1 & 0 \end{smallmatrix}\right) L_t \quad \text{otherwise}$$

and

$$L_t \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \left(\begin{smallmatrix} a & 0 \\ 0 & 1 \end{smallmatrix}\right) L_t \left(\begin{smallmatrix} 1 & 0 \\ c/a & 1 \end{smallmatrix}\right) \quad \text{if } a \neq 0 \qquad L_t \left(\begin{smallmatrix} 0 & b \\ c & d \end{smallmatrix}\right) = \left(\begin{smallmatrix} 0 & 1 \\ c & 0 \end{smallmatrix}\right) L_x \quad \text{otherwise.}$$

As before, certain products remain as normal forms, for example $L_x \left(\begin{smallmatrix} 1 & k \\ 0 & 1 \end{smallmatrix}\right)$. Such an operator acts on a function $f \in \mathcal{F}$ as $f(kt, t)$, collapsing the bivariate function f to the univariate restriction along the diagonal line $x = kt$.

The language of $\mathcal{R}$-operators is not very expressive, but enough for our modest purposes at this point—expressing the boundary problem (2) and computing its Green's operator. Let us first look at the general first-order boundary problem with constant coefficients, prescribing homogeneous "boundary values" on an arbitrary line. Fixing the parameters $a, b, c, k \in \mathbb{R}$, it reads as follows:

$$\boxed{\begin{array}{l} a\, u_x + b\, u_t = f \\ u(kt + c, t) = 0 \end{array}} \tag{6}$$

Here $(a,b)^t$ determines the direction (and speed) of the ground characteristics, while $x = kt + c$ gives the line of boundary values. Of course this excludes the horizontal lines $t = \text{const}$, which would have to be treated separately, in a completely analogous manner. Since (in this paper) we are interested only in regular boundary problems, the characteristics must have a transversal intersection with the line of boundary values. Hence we stipulate that $a - kb \neq 0$. Moreover, we will also assume $a \neq 0$; for otherwise one may switch the x- and t-coordinates. A straightforward computation (or a suitable computer algebra system) gives now

$$u(x,y) = \frac{1}{a} \int_X^x f(\xi, \tfrac{b}{a}(\xi - x) + t)\, d\xi \qquad \text{with} \qquad X = \frac{ac + (at - bx)k}{a - bk}.$$

This solution for the general case can be reduced to $(a,b)^t = (1,0)^t$ and $k = 0$ by first rotating (a,b) into horizontal position, then normalizing it through x-scaling, and finally shearing the line of boundary values into vertical position. This yields the factorization

$$u(x,y) = \left(\begin{smallmatrix} 1/K & -k/K \\ -b/L & a/L \end{smallmatrix}\right) \cdot \textstyle\int_{c/K}^x \cdot \left(\begin{smallmatrix} a & kL/K \\ b & L/K \end{smallmatrix}\right) f(x,y), \tag{7}$$

where $K = a - bk$ and $L = a^2 + b^2$. This is almost an $\mathcal{R}$-operator, except that we have only allowed $A_x = \int_0^x$ and its t-analog, so we cannot express $\int_{c/K}^x$ unless we allow more evaluations such that we could write the required integral as $A_x - L_x^{c/K} A_x$, where L_x^ξ acts on a function $g(x, y)$ as $g(\xi, y)$.

While it would be straightforward to incorporate such evaluations by adding suitable relations, it is enough for our purposes to restrict the line of boundaries: We require it to pass through the origin so that $c = 0$. In this case we have of course $\int_{c/K}^x = A_x$, and (7) shows that we can indeed write the Green's operator in the $\mathcal{R}$ language.

5 Implementation in Theorema

As explained in Sections 2 and 4, we compute the Green's operator of a boundary problem for an ODE as an *integro-differential operator*. These operators are realized as noncommutative polynomials (introduced by a generic construct for monoid algebras), taken modulo an infinite parametrized Gröbner basis.

As coefficients we allow either standard polynomials or—more generally—exponential polynomials. Informally speaking, an *exponential polynomial* is a linear combination of terms having the form $x^n e^{\lambda x}$, where n is a natural and λ a complex number. Both the standard and the exponential polynomials can again be generated as an instance of the monoid algebra, respectively using $\mathbb{N}$ and $\mathbb{N} \times \mathbb{C}$ as a term monoid. In this way, we have complete algorithmic control over the coefficient functions (modulo Mathematica's simplifier for constants); see also [22]. Alternatively, we can also take as coefficients all functions representable in Mathematica and let it do the operations on them.

We describe now briefly the representation of integro-differential operators and the implementation of the main algorithms solving, composing and factoring boundary problems. The implementation will soon be available at the website `www.theorema.org`. It is based on Theorema [6], a system designed as an integrated environment for doing mathematics, in particular proving, computing, and solving in various domains of mathematics. Its core language is higher-order predicate logic, containing a natural *programming language* such that algorithms can be coded and verified in a unified formal frame.

We make heavy use of *functors*, introduced and first implemented in Theorema by Buchberger. The general idea—and its use for structuring those domains in which Gröbner bases can be computed—is described in [23, 24], where one can also find references to original and early papers by Buchberger on the subject. For a general discussion of functor programming, see also [25].

Functors are a powerful tool for building up *hierarchical domains* in mathematics in a modular and generic way that unites elegance and formal clarity. In Theorema, the notion of a functor is akin to functors in ML, not to be confused with the functors of category theory. From a computational point of view, a Theorema functor is a higher-order function that produces a new domain (carrier and operations) from given domains: operations in the new domain are defined in terms of operations in the underlying domains. Apart from this computational aspect, functors also have an important reasoning aspect—a functor transports properties of the input domains to properties of the output domain, for example by "conservation theorems".

The `MonoidAlgebra` is the crucial functor that builds up *polynomials*, starting from the base categories of fields with an ordering and ordered monoids. We construct first the free vector space `V` over a field `K` generated by the set of words in an ordered monoid `W` via the functor `FreeVecSpc[K,W]`. Then we extend this domain by introducing a multiplication using the corresponding operations in `K` and `W` as follows.

```
MonoidAlgebra[K, W] = where[V = FreeVecSpc[K, W],
  Functor[P, any[c, d, f, g, ξ, η, m̄, n̄],
     𝕤 = ⟨⟩
     ...(* linear operations from V *)
     (* multiplication *)
     ⟨⟩ *_P g = ⟨⟩                                                         ]]
     f *_P ⟨⟩ = ⟨⟩
     ⟨⟨c, ξ⟩, m̄⟩ *_P ⟨⟨d, η⟩, n̄⟩ = ⟨⟨c *_K d, ξ *_W η⟩⟩ +_P ⟨⟨c, ξ⟩⟩ *_P ⟨n̄⟩ +_P ⟨m̄⟩ *_P ⟨⟨d, η⟩, n̄⟩
```

For building up the *integro-differential operators* over an integro-differential algebra $\mathcal{F}$ of coefficient functions, `FreeIntDiffOp[`$\mathcal{F}$`,K]` constructs an instance of the monoid algebra with the word monoid over the infinite alphabet consisting of the letters ∂ and $\int$ along with a basis of $\mathcal{F}$ and all multiplicative characters corresponding to evaluations at points in `K`.

```
Definition["IntDiffOp", any[ℱ, K],
 IntDiffOp[ℱ, K] = where[𝒜 = FreeIntDiffOp[ℱ, K], 𝒢 = GreenSystem[ℱ, K]
     QuotAlg[GBNF[𝒜, 𝒢]]]
]
```

The `GreenSystem` functor contains the encoding of the rewrite system described in Table 1 of [4, 14], representing a noncommutative Gröbner basis. The normal forms with respect to total reduction modulo infinite Gröbner bases are introduced in the `GBNF` functor, while the `QuotAlg` functor creates the quotient algebra from the corresponding canonical simplifier.

In Appendix A, we present a few examples of boundary problems for ODEs whose *Green's operators* are computed using (4), which now takes on the following concrete form in Theorema code.

```
GreensOp_P[F, ℬ] = (1_𝒜 -_𝒜 Proj_P[ℬ, F]) *_𝒜 RightInv_P[F]
```

Here $\mathcal{B}$ is the vector of boundary conditions and `F` the given fundamental system of solutions.

In a way similar to the integro-differential operators $\mathcal{F}[\partial, \int]$ for ODEs, we have also implemented the integro-differential operators $\mathcal{R}$ for the simple PDE setting outlined in Section 4. Using the same functor hierarchy, we added the corresponding rules for the operators D_x, D_t, A_x, A_t, L_x, L_t and the substitution operators defined by matrices in $\mathrm{GL}(\mathbb{R}, 2)$. Moreover, we implemented the computation of Green's operators for first-order boundary problems (7). With the factorization (5) we can then compute the Green's operator for the unbounded wave equation (Appendix A).

6 Conclusion

The implementation of our symbolic framework for boundary problems allows us in particular to *solve boundary problems* for *ODEs* from a given fundamental system of the corresponding homogeneous equations. Given a factorization of the differential operator and a fundamental system of one of the factors, we can also *factor boundary problems* into lower order problems. In both cases it would be interesting to investigate the combination with numerical approaches to differential equations and boundary problems. For example, how can we use a fundamental system coming from a numerical algorithm or how can numerical methods be adapted to deal with integral boundary conditions?

The current setting for *PDEs* is of course still very limited and should only be seen as a starting point for future work. But in combination with our factorization approach, we believe that it can be extended to include more complicated problems. For example, the wave equation on the *bounded interval* $[0, 1]$, which in our notation reads as $\mathcal{P} = (D_t^2 - D_x^2, [u(x, 0), u_t(x, 0), u(0, t), u(1, t)])$ with x ranging over $[0, 1]$ and t over $\mathbb{R}_{\geq 0}$, can be factored [5] into $\mathcal{P} = \mathcal{P}_1 \circ \mathcal{P}_2$ with

$$\mathcal{P}_1 = (D_t - D_x, [u(x, 0), \textstyle\int_{\max(1-t,0)}^1 u(\xi, \xi + t - 1)\, d\xi])$$

and $\mathcal{P}_2 = (D_t + D_x, [u(x, 0), u(0, t)])$. The more complicated structure of the Green's operator for $\mathcal{P}$ (it involves a finite sum with an upper bound depending on its argument) is reflected in the Green's operator for the left factor $\mathcal{P}_1$. Its computation leads in this case to a simple functional equation, but a systematic approach to compute and represent Green's operators for PDEs with *integral boundary conditions* still needs to be developed. In a generalized setting including the bounded wave equation, we would also have to allow for more complicated geometries: as a first step bounded intervals and then also arbitrary convex sets.

References

1. Stakgold, I.: Green's functions and boundary value problems. John Wiley & Sons, New York (1979)
2. Rosenkranz, M., Buchberger, B., Engl, H.W.: Solving linear boundary value problems via non-commutative Gröbner bases. Appl. Anal. **82** (2003) 655–675
3. Rosenkranz, M.: A new symbolic method for solving linear two-point boundary value problems on the level of operators. J. Symbolic Comput. **39**(2) (2005) 171–199
4. Rosenkranz, M., Regensburger, G.: Solving and factoring boundary problems for linear ordinary differential equations in differential algebras. J. Symbolic Comput. **43**(8) (2008) 515–544
5. Regensburger, G., Rosenkranz, M.: An algebraic foundation for factoring linear boundary problems. Ann. Mat. Pura Appl. (4) **188**(1) (2009) 123–151

6. Buchberger, B., Craciun, A., Jebelean, T., Kovacs, L., Kutsia, T., Nakagawa, K., Piroi, F., Popov, N., Robu, J., Rosenkranz, M., Windsteiger, W.: Theorema: Towards computer-aided mathematical theory exploration. J. Appl. Log. **4**(4) (2006) 359–652
7. Buchberger, B.: An algorithm for finding the bases elements of the residue class ring modulo a zero dimensional polynomial ideal (German). PhD thesis, Univ. of Innsbruck (1965) English translation J. Symbolic Comput. **41**(3-4) (2006) 475–511.
8. Buchberger, B.: Introduction to Gröbner bases. In Buchberger, B., Winkler, F., eds.: Gröbner bases and applications. Cambridge Univ. Press (1998)
9. Mora, T.: An introduction to commutative and noncommutative Gröbner bases. Theoret. Comput. Sci. **134**(1) (1994) 131–173
10. Köthe, G.: Topological vector spaces (Volume I). Springer, New York (1969)
11. Brown, R.C., Krall, A.M.: Ordinary differential operators under Stieltjes boundary conditions. Trans. Amer. Math. Soc. **198** (1974) 73–92
12. Kamke, E.: Differentialgleichungen. Lösungsmethoden und Lösungen. Teil I: Gewöhnliche Differentialgleichungen. Akademische Verlagsgesellschaft, Leipzig (1967)
13. Coddington, E.A., Levinson, N.: Theory of ordinary differential equations. McGraw-Hill Book Company, Inc., New York-Toronto-London (1955)
14. Rosenkranz, M., Regensburger, G.: Integro-differential polynomials and operators. In Jeffrey, D., ed.: Proceedings of ISSAC '08, New York, ACM (2008) 261–268
15. van der Put, M., Singer, M.F.: Galois theory of linear differential equations. Springer, Berlin (2003)
16. Schwarz, F.: A factorization algorithm for linear ordinary differential equations. In: Proceedings of ISSAC '89, New York, ACM (1989) 17–25
17. Tsarev, S.P.: An algorithm for complete enumeration of all factorizations of a linear ordinary differential operator. In: Proceedings of ISSAC '96, New York, ACM (1996) 226–231
18. Grigoriev, D., Schwarz, F.: Loewy- and primary decompositions of D-modules. Adv. in Appl. Math. **38** (2007) 526–541
19. Tsarev, S.P.: Factorization of linear partial differential operators and Darboux integrability of nonlinear PDEs. SIGSAM Bull. **32**(4) (1998) 21–28
20. Regensburger, G., Rosenkranz, M., Middeke, J.: A skew polynomial approach to integro-differential operators. In: Proceedings of ISSAC '09, New York, ACM (2009) to appear.
21. Cohn, P.M.: Further algebra and applications. Springer-Verlag, London (2003)
22. Buchberger, B., Regensburger, G., Rosenkranz, M., Tec, L.: General polynomial reduction with Theorema functors: Applications to integro-differential operators and polynomials. ACM Commun. Comput. Algebra **42**(3) (2008) 135–137
23. Buchberger, B.: Groebner rings and modules. In Maruster, S., Buchberger, B., Negru, V., Jebelean, T., eds.: Proceedings of SYNASC 2001. (2001) 22–25
24. Buchberger, B.: Groebner bases in Theorema using functors. In Faugere, J., Wang, D., eds.: Proceedings of SCC '08, LMIB Beihang University Press (2008) 1–15
25. Windsteiger, W.: Building up hierarchical mathematical domains using functors in Theorema. Electr. Notes Theor. Comput. Sci. **23**(3) (1999) 401–419

A Sample Computations

Let us again consider example (1). By our implementation, we obtain the Green's operator for the boundary problem with the corresponding Green's function. As noted in [3], the Green's function provides a canonical form for the Green's operator. In the following, we use the notation $Au = \int_0^x u(\xi)\, d\xi$, $Bu = \int_x^1 u(\xi)\, d\xi$, $Lu = u(0)$, $Ru = u(1)$, and $A1f(x,t) = \int_0^x f(\xi,t)\, d\xi$.

```
Compute[AsGreen_𝔤[GreensOp_𝔹[D², ⟨⟨⟨1, ⟨⟨"⌊⌋", 0⟩⟩⟩⟩, ⟨⟨1, ⟨⟨"⌊⌋", 1⟩⟩⟩⟩⟩]]]
```

```
-A x - x B + x A x + x B x
```

```
Compute[GreensFct_𝔤𝔣[GreensOp_𝔹[D², ⟨⟨⟨1, ⟨⟨"⌊⌋", 0⟩⟩⟩⟩, ⟨⟨1, ⟨⟨"⌊⌋", 1⟩⟩⟩⟩⟩]]]
```

$$\begin{cases} -\xi + x\,\xi & \Leftarrow \quad \xi \le x \\ -x + x\,\xi & \Leftarrow \quad x < \xi \end{cases}$$

As explained in Section 3, we can factor (1) along a factorization of the differential operator, given a fundamental system for the right factor. Here is how we can compute the boundary conditions of the left and right factor problems, respectively.

```
Compute[AsGreen_𝔤[Factorize_𝔹𝔪[D, D, ⟨⟨⟨1, ⟨⟨"⌊⌋", 0⟩⟩⟩⟩, ⟨⟨1, ⟨⟨"⌊⌋", 1⟩⟩⟩⟩⟩, ⟨⟨⟨1, ⟨⟩⟩⟩⟩]]]
```

```
⟨⟨A + B⟩, ⟨L⟩⟩
```

We consider as a second example the fourth order boundary problem [4, Ex. 33]:

$$\boxed{\begin{array}{l} u'''' + 4u = f, \\ u(0) = u(1) = u'(0) = u'(1) = 0. \end{array}} \tag{8}$$

Factoring the boundary problem along $D^4 + 4 = (D^2 - 2i)(D^2 + 2i)$, we obtain the following boundary conditions for the factor problems.

```
Compute[AsGreen_𝔤[Factorize_𝔹𝔪[D² - 2 ⅈ, D² + 2 ⅈ,
   ⟨⟨⟨1, ⟨⟨"⌊⌋", 0⟩⟩⟩⟩, ⟨⟨1, ⟨⟨"⌊⌋", 1⟩⟩⟩⟩, ⟨⟨1, ⟨⟨"⌊⌋", 0⟩, "∂"⟩⟩⟩, ⟨⟨1, ⟨⟨"⌊⌋", 1⟩, "∂"⟩⟩⟩⟩,
   ⟨⟨⟨1, ⟨⟨"⌈⌉", ⟨0, -1 + ⅈ⟩⟩⟩⟩⟩, ⟨⟨1, ⟨⟨"⌈⌉", ⟨0, 1 + (-1) ⅈ⟩⟩⟩⟩⟩⟩]]]
```

```
⟨⟨A ⅇ^(Complex[-1,1]) x + B ⅇ^(Complex[-1,1]) x, A ⅇ^(Complex[1,-1]) x + B ⅇ^(Complex[1,-1]) x⟩, ⟨L, R⟩⟩
```

With our implementation we can also compute its Green's operator and verify the solution presented in [4].

The final example for ODEs is a third order boundary problem with exponential coefficients.

$$\boxed{\begin{array}{l} u''' - (e^x + 2)u'' - u' + (e^x + 2)u = f, \\ u(0) = u(1) = u'(1) = 0. \end{array}} \tag{9}$$

Here we use as coefficient algebra all functions representable in Mathematica. The Green's operator is computed as follows.

```
Compute[GreensOp_𝔹[⟨⟨⟨1, mma[ⅇ^x]⟩⟩, ⟨⟨1, mma[ⅇ^-x]⟩⟩, ⟨⟨-1, mma[ⅇ^ⅇ^x ⅇ^-x]⟩, ⟨1, mma[ⅇ^ⅇ^x]⟩⟩⟩,
  ⟨⟨⟨1, ⟨⟨"⌊⌋", 0⟩⟩⟩⟩, ⟨⟨1, ⟨⟨"⌊⌋", 1⟩⟩⟩⟩, ⟨⟨1, ⟨⟨"⌊⌋", 1⟩, "∂"⟩⟩⟩⟩]]
```

$$
\begin{aligned}
&(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}} A + (-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}} B + (-1)\,(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}+(-1)\,x} A + \\
&(-1)\,(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}+(-1)\,x} B + \left(\frac{1}{2} + \frac{1}{2}\,(-1+e)^{-2}\right) e^{-1 x} A + \frac{1}{2}\,(-1+e)^{-2}\, e^{-1 x} B + \\
&\left(\frac{-1}{2}\right)(-1+e)^{-2}\, e^{x} A + \left(\frac{-1}{2}\right)(-1+e)^{-2}\, e^{x} B + (-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}} A\, e^{-2 x} + \\
&(-2)\,(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}} A\, e^{-1 x} + (-1)\, e^{e^{x}} B\, e^{-1 e^{x}+(-2)\,x} + (-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}} B\, e^{-2 x} + \\
&(-2)\,(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}} B\, e^{-1 x} + (-1)\,(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}+(-1)\,x} A\, e^{-2 x} + 2\,(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}+(-1)\,x} A\, e^{-1 x} + \\
&e^{e^{x}+(-1)\,x} B\, e^{-1 e^{x}+(-2)\,x} + (-1)\,(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}+(-1)\,x} B\, e^{-2 x} + 2\,(-1+e)^{-2}\, e^{-1 e}\, e^{e^{x}+(-1)\,x} B\, e^{-1 x} + \\
&\left(1 + \frac{1}{2}\left(-1 + (-1+e)^{-2}\right)\right) e^{-1 x} A\, e^{-2 x} + \left(-1 + (-1)\,(-1+e)^{-2}\right) e^{-1 x} A\, e^{-1 x} + \\
&\frac{1}{2}\left(-1 + (-1+e)^{-2}\right) e^{-1 x} B\, e^{-2 x} + (-1)\,(-1+e)^{-2}\, e^{-1 x} B\, e^{-1 x} + \left(\frac{-1}{2} + \frac{1}{2}\,(-2+e)\,(-1+e)^{-2}\, e\right) e^{x} A\, e^{-2 x} + \\
&(-1+e)^{-2}\, e^{x} A\, e^{-1 x} + \frac{1}{2}\,(-2+e)\,(-1+e)^{-2}\, e\, e^{x} B\, e^{-2 x} + (-1+e)^{-2}\, e^{x} B\, e^{-1 x}
\end{aligned}
$$

As a last example, we return to the boundary problem for the *wave equation* (2). With Proposition 1 and using the factorization (5), we can compute the Green's operator for (2) simply by composing the Green's operators of the first-order problems $\mathcal{P}_1 = (D_t - D_x, [u(x,0)])$ and $\mathcal{P}_2 = (D_t + D_x, [u(x,0)])$. Relative to the setting in Section 4, we switch the x- and t-coordinates.

```
Compute[GreensOp_𝔹[1, -1, 0] ⋆_𝔹 GreensOp_𝔹[1, 1, 0]]
```

```
⟨⟨1, ⟨⟨mat, ⟨⟨1, 0⟩, ⟨-1, 1⟩⟩⟩, A1, ⟨mat, ⟨⟨1, 0⟩, ⟨2, 1⟩⟩⟩, A1, ⟨mat, ⟨⟨1, 0⟩, ⟨-1, 1⟩⟩⟩⟩⟩⟩
```

Interchanging again t and x, this corresponds in the usual notation to $G_1 f(x,t) = \int_0^t f(\xi, -\xi + x + t)\, d\xi$ and $G_2 f(x,t) = \int_0^t f(\xi, \xi + x - t)\, d\xi$, which yields

$$G_2 G_1 f(x,t) = \int_0^t \int_0^\tau f(\xi, 2\tau - \xi + x - t)\, d\xi\, d\tau$$

for the Green's operator of the unbounded wave equation (2).