Symbolic Computation for Ordinary Boundary Problems in Maple

Anja Korporal
Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, 4040 Linz, Austria
`Anja.Korporal@oeaw.ac.at`

Georg Regensburger*
INRIA Saclay – Île de France, Project DISCO, L2S, Supélec, 91192 Gif-sur-Yvette Cedex, France
`Georg.Regensburger@ricam.oeaw.ac.at`

Markus Rosenkranz
University of Kent, Cornwallis Building, Canterbury, Kent CT27NF, United Kingdom
`M.Rosenkranz@kent.ac.uk`

We present a new version of the Maple package *IntDiffOp* [6, 7] for Symbolic Computation with boundary problems for linear ordinary differential equations. The solution of boundary problems for linear ordinary differential equations is of great practical importance, and there is a vast literature on their analytic treatment [4, 14, 1, 5, 2]. A new symbolic approach was introduced in [10] and subsequently generalized to a differential algebra setting in [11]. For a recent survey and references on our symbolic approach to boundary problems we refer to [13]. The first implementations were coded in Mathematica/TH∃OREM∀, as an external package in [10] for boundary problems with constant coefficients and as an internal functor in [12, 15, 13] for generic integro-differential algebras. In contrast to the stepwise reduction approach of the Mathematica packages, the *IntDiffOp* package uses normal forms (up to basis expansion) [9]. In [7] we give a detailed description of the functionality for solving and factoring regular boundary problems (i.e. those having a unique solution for every right hand side), similar to the package in [15]. Moreover, we introduce an algorithmic approach for singular boundary problems and generalized Green's operators [8, 3].

In the new version, our main purpose was on the one hand to improve the usability of the procedures for treating boundary problems. The functions accept, where applicable, input in usual Maple syntax as used for *dsolve*. So one can solve and manipulate boundary problems with the *IntDiffOp* package without having to use the syntax for integro-differential operators. On the other hand, we provide an interface for integro-differential operators by overloading the usual arithmetic operators. The package is available at `http://www.risc.jku.at/people/akorpora/index.html`. We also provide two worksheets illustrating the new functionalities.

The new command for solving boundary problems is *bdsolve*. It uses the Maple function *dsolve* for computing a fundamental system for the homogeneous differential equation. Boundary problems are represented as a list consisting of a differential equation as the first entry followed by (possibly inhomogeneous) boundary conditions. If the optional argument *gf* is set to 1, the Green's function for regular two-point boundary problems with homogeneous boundary conditions is computed. The function *bdsolve* can also solve multipoint boundary problems.

```
> with(IntDiffOp):
> ode := diff(u(x), x, x) = f(x): bcs := u(0) = 2, u(1) = 5:
> bdsolve([ode, bcs]);
```

$$x\left(\int_0^x f(x)\,\mathrm{d}x\right)-\left(\int_0^x xf(x)\,\mathrm{d}x\right)-x\left(\int_0^1 f(x)\,\mathrm{d}x\right)+x\left(\int_0^1 xf(x)\,\mathrm{d}x\right)+2+3x$$

```
> bdsolve([ode, bcs], gf = 1);
```

$$\begin{cases}-\xi+x\xi & 0\le\xi \text{ and } \xi\le x \text{ and } x\le 1\\ -x+x\xi & 0\le x \text{ and } x\le\xi \text{ and } \xi\le 1\end{cases}$$

*Supported by the Austrian Science Fund (FWF): J 3030-N18.

Maple's *dsolve* can currently compute solutions for simple boundary problems, but there is no systematic support for boundary conditions. If in the previous example the first condition is replaced by $u(0) + \mathrm{D}(u)(0) = 2$, it does not find a solution. Moreover, the functions of the *IntDiffOp* package allow global conditions involving integrals, which occur naturally during factoring boundary problems or treating singular problems. We also use normal forms for integro-differential operators, involving only single integrals, while *dsolve* returns nested integrals.

For singular boundary problems, which are not solvable for all forcing functions f, the command *CompatibilityConditions* computes the constraints for f. For such problems, one has to choose a complement of the image of the differential operator restricted to the functions satisfying the boundary conditions. The solution is then computed for the projection of f onto the space of functions satisfying the compatibility conditions along this exceptional space. In *bdsolve*, the exceptional space can be specified with the optional argument *es*.

```
> bcs3 := D(u)(0) = 0, D(u)(1) = 0, u(1)=0:
> CompatibilityConditions([ode, bcs3]);
```

$$\int_0^x f(x)\,\mathrm{d}x = 0$$

```
> bdsolve([ode, bcs3], es=1);
```

$$x\left(\int_0^x f(x)\,\mathrm{d}x\right) - \left(\int_0^x x f(x)\,\mathrm{d}x\right) - \frac{1}{2}\left(\int_0^1 f(x)\,\mathrm{d}x\right)x^2 - \frac{1}{2}\int_0^1 f(x)\,\mathrm{d}x + \int_0^1 x f(x)\,\mathrm{d}x$$

The procedure *IsRegular* uses the same input as *bdsolve* and tests if a boundary problem is regular and if a set of functions forms a basis for an admissible exceptional space. Also other functions from the *IntDiffOp* package like *GreensOperator* or *FundamentalSystem* have been modified to allow input in Maple notation. The new function *bdfactor* factors boundary problems into lower-order problems. It uses the Maple function *dfactor* for factoring differential operators. The output of *bdfactor* are boundary problems in Maple notation, which can also serve as input for further computations.

```
> bcs2 := u(0) = 0, u(1) = 0:
> G := GreensOperator([ode, bcs2]):
> f1, f2 := bdfactor([ode, bcs2]);
```

$$f1, f2 := \left[\frac{\mathrm{d}}{\mathrm{d}x}u(x) = f(x), \int_0^1 u(x)\,\mathrm{d}x = 0\right], \left[\frac{\mathrm{d}}{\mathrm{d}x}u(x) = f(x), u(0) = 0\right]$$

```
> G1 := GreensOperator(f1): G2:= GreensOperator(f2):
> SubtractOperator(G, MultiplyOperator(G2, G1));
```

$$0$$

The *IntDiffOperations* package predefines basic integro-differential operators and overloads the arithmetic operators. The differential and integral operator are denoted by d and a. For the output we use the symbols defined in *IntDiffOp* (per default D and A). The action of integro-differential operators on functions is denoted by `` `&*` ``. Algebraic expressions are interpreted as multiplication operators, and the evaluation at a point c is denoted by *e(c)*.

```
> with(IntDiffOperations):
> d, a, d&*f(x), a&*f(x);
```

$$\mathrm{D}, A, \frac{\mathrm{d}}{\mathrm{d}x}f(x), \int_0^x f(x)\,\mathrm{d}x$$

```
> (1+x^2+sin(x))&*f(x);
```

$$\left(1 + x^2 + \sin(x)\right) f(x)$$

```
>  e(0), e(1), e(0)&*f(x), e(1)&*f(x);
```

$$E[0],\ \ E[1],\ \ f(0),\ \ f(1)$$

For the noncommutative multiplication of integro-differential operators we overload `‘.‘`. The first examples below show the defining relations for the algebra of integro-differential operators with polynomial coefficients. We overload respectively `‘+‘`, `‘-‘`, `‘^‘`, and `‘*‘` for the sum, difference, exponentiation, and scalar multiplication.

```
> d.x, d.a, a.d, a.a, e(1).e(0);
```

$$1 \;+\; x \;.\; D,\; 1,\; 1 \;-\; E[0],\; x \;.\; A \;-\; A \;.\; x,\; E[0]$$

```
> 1+3*d + (x^2).d+sin(x).d^2-a^2+(x^2).e(0)+e(1).(a.x);
```

$$1 \;+\; (x\hat{\ }2 \;+\; 3) \;.\; D \;+\; sin(x) \;.\; D\hat{\ }2 \;-\; x \;.\; A \;+\; A \;.\; x \;+\; x\hat{\ }2 \;.\; E[0] \;+\; E[1] \;.\; A \;.\; x$$

```
> %&*f(x);
```

$$f(x) + (x^2+3)\Big(\frac{\mathrm{d}}{\mathrm{d}x} f(x)\Big) + \sin(x)\Big(\frac{\mathrm{d}^2}{\mathrm{d}x^2} f(x)\Big) - x\Big(\int_0^x f(x)\,\mathrm{d}x\Big) + \int_0^x x f(x)\,\mathrm{d}x + x^2 f(0) + \int_0^1 x f(x)\,\mathrm{d}x$$

References

[1] R. P. Agarwal. *Boundary value problems for higher order differential equations.* World Scientific Publishing Co. Inc., Teaneck, NJ, 1986.

[2] R. P. Agarwal and D. O’Regan. *An introduction to ordinary differential equations.* Springer, New York, 2008.

[3] A. Ben-Israel and T. N. E. Greville. *Generalized inverses.* Springer-Verlag, New York, second edition, 2003.

[4] E. A. Coddington and N. Levinson. *Theory of ordinary differential equations.* McGraw-Hill Book Company, Inc., New York-Toronto-London, 1955.

[5] D. G. Duffy. *Green’s functions with applications.* Chapman & Hall/CRC, Boca Raton, FL, 2001.

[6] A. Korporal, G. Regensburger, and M. Rosenkranz. A Maple package for integro-differential operators and boundary problems. *ACM Commun. Comput. Algebra*, 44(3):120–122, 2010. Poster at ISSAC ’10.

[7] A. Korporal, G. Regensburger, and M. Rosenkranz. Regular and singular boundary problems in Maple. In V. Gerdt, W. Koepf, E. Mayr, and E. Vorozhtsov, editors, *Proceedings of CASC 2011 (Computer Algebra in Scientific Computing)*, volume 6885 of *LNCS*, pages 280–293, Berlin / Heidelberg, 2011. Springer.

[8] W. S. Loud. Some examples of generalized Green’s functions and generalized Green’s matrices. *SIAM Rev.*, 12:194–210, 1970.

[9] G. Regensburger, M. Rosenkranz, and J. Middeke. A skew polynomial approach to integro-differential operators. In J. P. May, editor, *Proceedings of ISSAC ’09*, pages 287–294, New York, NY, USA, 2009. ACM.

[10] M. Rosenkranz. A new symbolic method for solving linear two-point boundary value problems on the level of operators. *J. Symbolic Comput.*, 39(2):171–199, 2005.

[11] M. Rosenkranz and G. Regensburger. Solving and factoring boundary problems for linear ordinary differential equations in differential algebras. *J. Symbolic Comput.*, 43(8):515–544, 2008.

[12] M. Rosenkranz, G. Regensburger, L. Tec, and B. Buchberger. A symbolic framework for operations on linear boundary problems. In V. P. Gerdt, E. W. Mayr, and E. H. Vorozhtsov, editors, *Proceedings of CASC 2009 (Computer Algebra in Scientific Computing)*, volume 5743 of *LNCS*, pages 269–283, Berlin, 2009. Springer.

[13] M. Rosenkranz, G. Regensburger, L. Tec, and B. Buchberger. Symbolic analysis for boundary problems: From rewriting to parametrized Gröbner bases. In U. Langer and P. Paule, editors, *Numerical and Symbolic Scientific Computing: Progress and Prospects*, pages 273–331. SpringerWienNew York, Vienna, 2012.

[14] I. Stakgold. *Green’s functions and boundary value problems.* John Wiley & Sons, New York, 1979.

[15] L. Tec. *A Symbolic Framework for General Polynomial Domains in Theorema: Applications to Boundary Problems.* PhD thesis, RISC, University of Linz, 2011.