Turnkey Solutions to PDEs in Exact Real Computation

Svetlana Selivanova* and Martin Ziegler

School of Computing, KAIST

Keywords: Exact Real Computation; Guaranteed Precision; Computable Analysis; PDEs; Difference Schemes; Analytic Solutions

Introduction

We turn the rigorous but theoretical approach to computing with continuous data [8] into practice, complementing classical numerical and analytic methods for solving broad classes of initial-value and boundary-value problems (IVP and BVP) for partial differential equations (PDEs). The *Exact Real Computation* paradigm[1] allows to conveniently implement imperative algorithms involving real numbers, converging sequences, and smooth functions without the hassles of Turing machines. This approach differs from traditional Reliable Numerics in considering real numbers as exact entities (as opposed to intervals [5]) while guaranteeing output approximations up to error $1/2^n$ (as opposed to intermediate precision propagation), where n is the output error parameter. We develop a turnkey solver, including careful calculations of internal parameters (such as spatial grid and time step size) and in agreement with complexity predictions [4], in dependence on n. This is the starting point towards actual implementation.

Difference Schemes in Exact Real Computation

Consider IVP and BVP for systems of PDEs of the form

$$\begin{cases} \mathbf{u}_t = L\mathbf{u} + f(t,x) \in C^p(\Omega, \mathbb{R}^n), \\ \mathbf{u} \mid_{t=0} = \varphi(x) \in C^q(\Omega, \mathbb{R}^n), \\ \left(\mathcal{L}\mathbf{u} \mid_{\partial\Omega\times[0,T]} = \psi(y)\right). \end{cases} \tag{1}$$

Here $\partial\Omega$ is the boundary of the compact set $\Omega \subset \mathbb{R}^m$, $x \in \Omega$, $y \in \partial\Omega \times [0,T]$, $L = \sum\limits_{|\alpha|\le s} A_\alpha(x,u)\frac{\partial^\alpha}{\partial x^\alpha}$. For a boundary-value problem (the Cauchy problem being stated without the last condition in the parentheses of (1)), $\mathcal{L}$ is a linear operator.

Suppose the given IVP and BVP be *well posed* in that the classical solution $\vec{u} : [0;1] \times \overline{\Omega} \to \mathbb{R}$ (i) exists, (ii) is unique, and (iii) depends continuously on φ. More precisely we assume that $u(t,x) \in C^2$ and its C^2-norm is bounded linearly by C^2-norms of the data as $||u||_{C^2} \le c_{\mathrm{u}}||\varphi||_{C^2}$ (in functional spaces guaranteeing all the required properties). Moreover suppose that the given IVP and BVP admit a (iv) stable (with stability coefficient c_{st}) and (v) approximating with at least the first order of accuracy (and approximation coefficient c_{app}) explicit *difference scheme* [2].

Then taking any (binary-rational) uniform space grid step h such that

$$h \le 1/\left(c_{\mathrm{u}} \cdot ||D_x^2\varphi|| \cdot (1 + c_{\mathrm{st}} \cdot c_{\mathrm{app}}) \cdot 2^n\right)$$

and (binary-rational) time step τ meeting the Courant inequatlity $\tau \le \nu h$, we can apply the standard (explicit) difference scheme iterations, treating all coefficients as exact reals. In this way we get an approximation to the solution with the precision $1/2^n$.

The coefficients c_{u}, c_{st} and c_{app} were explicitly expressed in [6] via (derivatives of) φ and $A_\alpha = A_\alpha^* = \mathrm{const}$ for a particular difference scheme for symmetric hyperbolic systems.

*Corresponding author.

[1]See arXiv:1608.05787v4

For the case when $A_\alpha(x, u) = A_\alpha(x)$ and $f(t, x) = 0$ it is possible to improve the bit cost of thus obtained algorithm by applying efficient matrix powering instead of step-by-step iterations [4].

Analytic PDEs in Exact Real Computation

For IVP with *analytic* A_α, f and φ in (1) we can rigorously compute solutions using analytic series, treating their coefficients as exact reals and applying iterations of [1], §4.6.3.

In the linear case $A_\alpha = A_\alpha(x)$, $f = 0$, it is more efficient to use the exponentiation series $\mathbf{u}(t, x) = \exp(tL)\varphi(x) = \sum_K t^K/K! \cdot L^K\varphi(x)$ and recursive operator powering, as suggested in [4]. More precisely, the n-th term of this power series gives approximation of the solution with precision $1/2^n$, provided that conditions of Theorem 8 of [4] hold.

Conclusion

For analytic PDEs we develop the series technique in addition to the possible application of difference schemes, because, as proved in [4] for the linear case (Theorem 3), it yields PTIME complexity bounds provided the input is PTIME computable. For the difference scheme approach the best complexity bound which we were so far able to establish for the linear case and PTIME inputs, was PSPACE (for particular examples $\#P^{\#P}$), i.e. much "worse" than PTIME.

Note also that finding the solution to the 2-dimensional Poisson equation was proved in [3] to be optimally in $\#P$ while solutions to Navier-Stokes equations were proved in [7] to be computable, but the proofs do not provide explicit algorithms.

Acknowledgement

Supported by the National Research Foundation of Korea (grant NRF-2017R1E1A1A03071032), and by the Korean Ministry of Science and ICT (grant NRF-2016K1A3A7A03950702.

References

[1] L. Evans. Partial differential equations. Graduate Studies in Mathematics, v. 19, American Mathematical Society, 1998.

[2] S. Godunov and V. Ryaben'kii. *Difference Schemes: An Introduction to the Underlying Theory*. Elsevier Science Ltd, 1987.

[3] A. Kawamura, F. Steinberg, and M. Ziegler. On the computational complexity of the Dirichlet problem for Poisson's equation. Mathematical Structures in Computer Science 27:8, P. 1437–1465, 2017.

[4] I. Koswara, S. Selivanova, and M. Ziegler. Computational complexity of real powering and improved solving linear differential equations. In *Proc. 14th International Computer Science Symposium in Russia*, volume 11532 of *LNCS*, 2019.

[5] M. Plum. Computer-assisted proofs for semilinear elliptic boundary value problems. Japan J. Indust. Appl. Math., V. 26, No 2-3, P. 419-442., 2009.

[6] S. Selivanova and V. Selivanov. Computing solution operators of boundary-value problems for some linear hyperbolic systems of PDEs. Logical Methods in Computer Science, 13 (4), 2017.

[7] S. M. Sun, N. Zhong, and M. Ziegler. On computability of Navier-Stokes' equation. volume 9136 of *LNCS*, 2015.

[8] K. Weihrauch. Computable analysis. Springer, Berlin, 2000.