Rigorous computation of solutions to differential equations problems, and algorithmic complexity

Svetlana Selivanova

Mathematical Center in Akademgorodok

WDCM-2022, October 29

Dedicated to Victor Selivanov's 70th Birthday

Plan

I. Motivational examples

II. Computability of solutions to differential equations and related problems

III. Complexity classification of differential equations, in comparison to other classifications

IV. Conclusion

I. Motivational examples

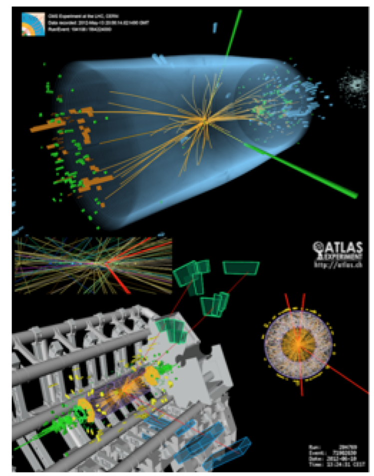
1) **Particle tracking in electromagnetic fields** (2022 in progress, j.w.w. Yannis K. Semertzidis and Holger Thies)

For particle velocity $\vec{\beta}$ and rest spin $\vec{s}$ in external fields $\vec{E}$, $\vec{B}$

$$\frac{d\vec{\beta}}{dt} = \frac{e}{m\gamma c}\left[\vec{E} + c\vec{\beta} \times \vec{B} - \vec{\beta}(\vec{\beta} \cdot \vec{E})\right]$$

$$\frac{d\vec{s}}{dt} = \frac{e}{m}\vec{s} \times [\left(a + \frac{1}{\gamma}\right)\vec{B} - \frac{a\gamma}{\gamma+1}\vec{\beta}(\vec{\beta} \cdot \vec{B}) -$$

$$- \left(a + \frac{1}{\gamma+1}\right)\frac{\vec{\beta} \times \vec{E}}{c}]$$



Candidate Higgs boson events from collisions between protons. Image credit: CERN Document Server

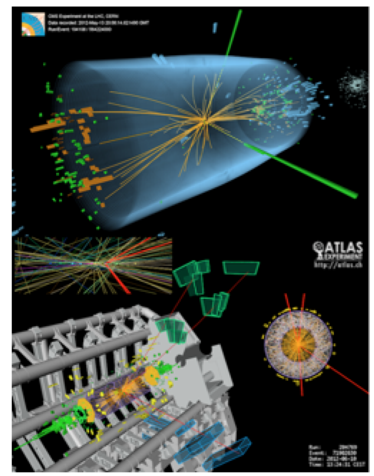
I. Motivational examples

1) **Particle tracking in electromagnetic fields** (2022 in progress, j.w.w. Yannis K. Semertzidis and Holger Thies)

For particle velocity $\vec{\beta}$ and rest spin $\vec{s}$ in external fields $\vec{E}$, $\vec{B}$

$$\frac{d\vec{\beta}}{dt} = \frac{e}{m\gamma c}\left[\vec{E} + c\vec{\beta} \times \vec{B} - \vec{\beta}(\vec{\beta} \cdot \vec{E})\right]$$

$$\frac{d\vec{s}}{dt} = \frac{e}{m}\vec{s}\times[\left(a + \frac{1}{\gamma}\right)\vec{B} - \frac{a\gamma}{\gamma+1}\vec{\beta}(\vec{\beta}\cdot\vec{B}) - \left(a + \frac{1}{\gamma+1}\right)\frac{\vec{\beta}\times\vec{E}}{c}]$$



Candidate Higgs boson events from collisions between protons. Image credit: CERN Document Server

Issue: Very small scale, *high precision* needed (otherwise, violation of energy conservation law in the simulation)

2) **Porous media** (2022 in progress, j.w.w. Yalchin Efendiev, Viet Ha Hoang and Holger Thies)

Multiscale problem

$$-\frac{\partial}{\partial x_i}(\kappa_{ij}^{\epsilon}(x)\frac{\partial}{\partial x_j}u) = f$$

with some boundary conditions;

$$\kappa_{ij}^{\epsilon}(x) = \kappa_{ij}(x, \frac{x}{\epsilon}).$$

2) **Porous media** (2022 in progress, j.w.w. Yalchin Efendiev, Viet Ha Hoang and Holger Thies)

Multiscale problem

$$-\frac{\partial}{\partial x_i}(\kappa^{\epsilon}_{ij}(x)\frac{\partial}{\partial x_j}u) = f$$

with some boundary conditions;

$$\kappa^{\epsilon}_{ij}(x) = \kappa_{ij}(x, \frac{x}{\epsilon}).$$

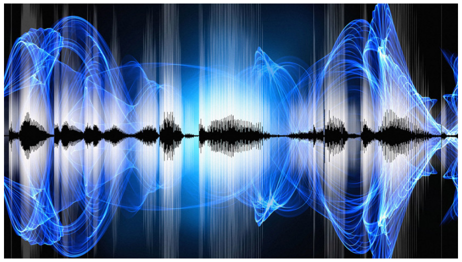
Issue: Compute the solution

$$u = u_0(x) + \epsilon u_1(x, \frac{x}{\epsilon}) + ...$$

with *guaranteed precision* for arbitrarily small ε

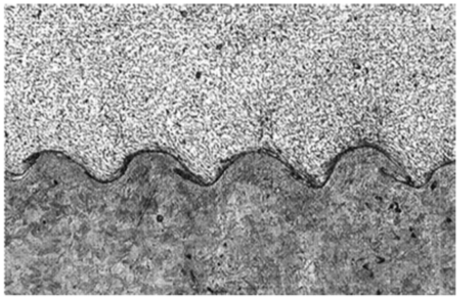
3) Symmetric hyperbolic systems (2007, j.w.w. Sergei K. Godunov)

$$\begin{cases} A(x)\frac{\partial}{\partial t}\vec{u} = \sum_{i=1}^{m} B_i(x)\frac{\partial}{\partial x_i}\vec{u}, \\ \vec{u}(0) = \varphi(x). \end{cases}$$



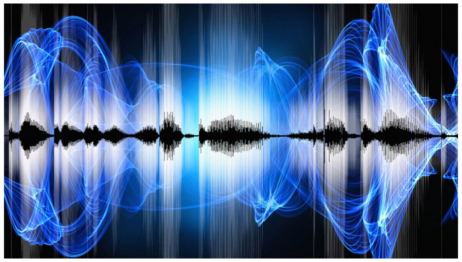
$$A = A^* > 0, \;\; B_i = B_i^*$$

Examples: acoustics, elasticity, electromagnetism



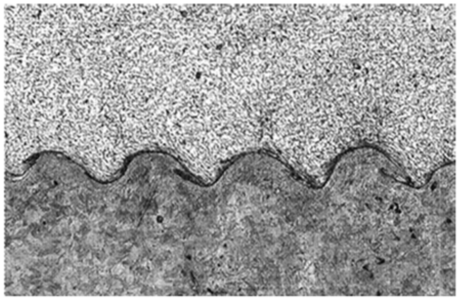
3) Symmetric hyperbolic systems (2007, j.w.w. Sergei K. Godunov)

$$\begin{cases} A(x)\frac{\partial}{\partial t}\vec{u} = \sum_{i=1}^{m} B_i(x)\frac{\partial}{\partial x_i}\vec{u}, \\ \vec{u}(0) = \varphi(x). \end{cases}$$



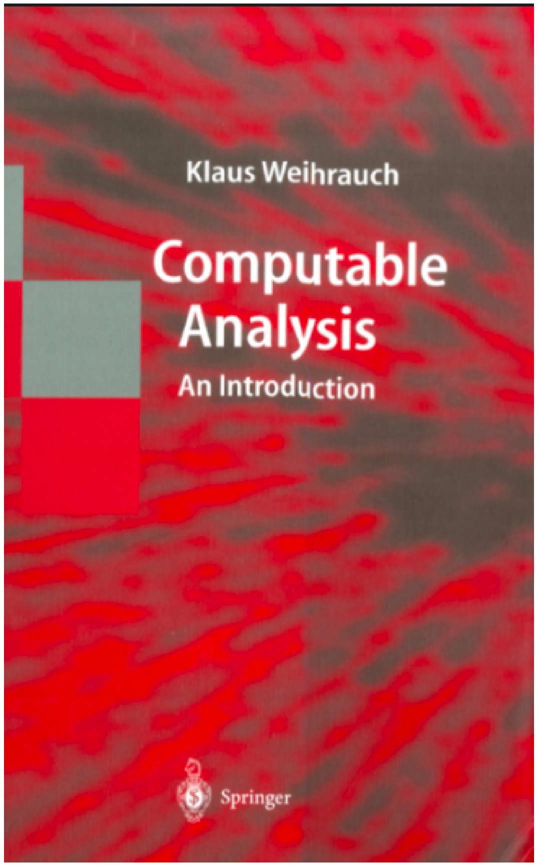
$$A = A^* > 0,\ \ B_i = B_i^*$$

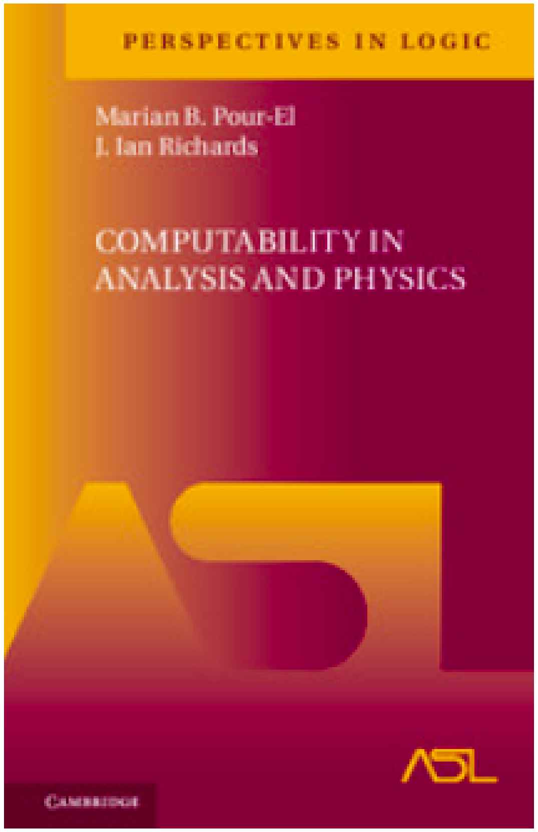
Examples: acoustics, elasticity, electromagnetism



Issue: Compute the solution $\vec{u}$ with *guaranteed precision*. Any convergent difference method involves computing *eigenvectors* of matrix pencils $\lambda A - B$, which is **discontinuous** (thus, not computable)

II. Computability of solutions to differential equations and related problems





II. Computability of solutions to differential equations and related problems

Symmetric hyperbolic systems (j.w.w. Victor Selivanov 2008; final journal version 2017)

$$\begin{cases} A\frac{\partial \mathbf{u}}{\partial t} + \sum\limits_{i=1}^{m} B_i \frac{\partial \mathbf{u}}{\partial x_i} = f(t,x), \ t \geq 0, \\ \mathbf{u}|_{t=0} = \varphi(x_1,\ldots,x_m). \end{cases}$$

Here $A = A^* > 0$ and $B_i = B_i^*$ are constant symmetric $n \times n$-matrices, $t \geq 0$, $x = (x_1,\ldots,x_m) \in Q = [0,1]^m$.

We study: computability properties of the solution operator

$$R : (A, B_i, f, \varphi) \mapsto \mathbf{u}$$

▷ **Symmetric Hyperbolic systems: computability questions**

We study: computability properties of the solution operator

$$R : (A, B_i, f, \varphi) \mapsto \mathbf{u}$$

Questions:

- Is the solution operator **computable** from f, φ? In which classes of functions? $||\varphi - \varphi^j|| < 2^{-j}$, $||f - f^j|| < 2^{-j} \mapsto ||\mathbf{u} - u^j|| < 2^{-n}$

- From the matrix coefficients A, B_i? $||A - A^j|| < 2^{-j}$, $||B_i - B_i^j|| < 2^{-j}$, $||\varphi - \varphi^j|| < 2^{-j}$, $||f - f^j|| < 2^{-j} \mapsto ||\mathbf{u} - u^n|| < 2^{-n}$

Results (joint work with V. Selivanov) about computability (2017) and complexity (2018,2021)

I. The solution operator

$$(\varphi, f) \mapsto \mathbf{u}$$

of (1), (2) is computable provided that the first and second partial derivatives of φ, f are uniformly bounded. Input: in sup-norm, output: in L_2-norm.

II. 1) The operator of the domain of existence and uniqueness $(A, B_1, \ldots, B_m) \mapsto H$ is computable (H is an intersection of $t \geq 0$, $x_i - \lambda_{\max}^{(i)} t \geq 0$, $x_i - 1 - \lambda_{\min}^{(i)} t \leq 0$, $(i = 1, \ldots, m)$, where $\{\lambda_k^{(i)}\}_{k=1}^n$ are the **eigenvalues** of $A^{-1}B_i$. Assume $\lambda_{\min}^{(i)} < 0 < \lambda_{\max}^{(i)}$ for all $i = 1, \ldots, m$.);

2) The solution operator

$$(\varphi, f, A, B_1, \ldots, B_m, n_A, n_1, \ldots, n_m) \mapsto \mathbf{u}$$

of (1), (2) is computable under certain additional spectral conditions on A, B_i.

Here n_A is the cardinality of spectrum of A (i.e. the number of different eigenvalues); n_i are the cardinalities of spectra of the matrix pencils $\lambda A - B_i$.

Eigenvectors are in general not computable! (Ziegler, Brattka)

3) The solution operator $(\varphi, f, A, B_1, \ldots, B_m) \mapsto \mathbf{u}$ of (1), (2) is computable when the coefficients of A, B_i run through an arbitrary computable real closed subfield of $\mathbb{R}$ (e.g. the set $\mathbb{A}$ of algebraic reals, or the real closure of $\mathbb{A} \cup \{c_1, \ldots, c_p\}$, $c_j \in \mathbb{R}_c$).

$\triangleright$ **Bit-complexity for coefficients from $\mathbb{A}$: EXP or PTIME?**

Theorem. **PTIME** if (for $A, B_i(\mathbb{A})$):

m, n, a, M are fixed positive integers;

the quantities $||A||_2$, $\frac{\lambda_{max}(A)}{\lambda_{min}(A)}$,

$$\max_i \Big\{ ||B_i||_2, ||(A^{-1}B_i)^2||_2, \max_k \{|\mu_k| : \det(\mu_k A - B_i) = 0\}, \sup_{t,x} ||\frac{\partial^2 f}{\partial x_i \partial t}(t,x)||_2 \Big\},$$

and

$$\max_{i,j} \Big\{ ||A^{-1}B_iA^{-1}B_j - A^{-1}B_jA^{-1}B_i||_2, \sup_{t,x} ||\frac{\partial^2 f}{\partial x_i \partial x_j}(t,x)||_2, \sup_x ||\frac{\partial^2 \varphi}{\partial x_i \partial x_j}(x)||_2 \Big\}$$

are bounded by M.

$\triangleright$ Bit-complexity for coefficients from $\mathbb{A}$: EXP or PTIME?

The proof heavily relies on:

• - using a stable difference scheme approximating (1), (2) and results on its convergence;

-proofs of the existence and uniqueness theorems for (1);

-properties of multilinear interpolations.

• deep results of computer algebra for polynomial arithmetic and computations in the fields of algebraic reals due to Loos, Collins, Grigoriev etc. and those recently considered by Alaev and Selivanov (including PTIME-presentability of $\mathbb{A}$ and PTIME computability of root finding).

▷ **Bit-complexity for coefficients from $\mathbb{A}$: EXP or PTIME?**

• Polynomial-time computability (in some fields of algebraic reals) of finding eigenvectors of matrix pencils $\lambda A - B_i$ (recall that this problem is not computable in the field of reals). In particular, this is crucial for finding in polynomial time steps h, τ guaranteeing the stability of the difference scheme.

• Our proof is a mix of methods typical for symbolic and numerical computations.

• Our methods apply only to algebraic matrices because it is currently open whether there is a PTIME-presentable real closed field of reals which contains a transcendental number.

▷ **Primitive recursiveness of the solution operator**

T h e o r e m. Let $M, p \geq 2$ be integers. Then the solution operator $(A, B_1, \ldots, B_m, \varphi) \mapsto \mathbf{u}$ for (1) is a PR-computable function (uniformly on m, n) from $S_+ \times S^m \times C_s^{p+1}(Q, \mathbb{R}^n)$ to $C^p_{sL_2}(H, \mathbb{R}^n)$ where S and S^+ are respectively the sets of all symmetric and symmetric positively definite matrices from $M_n(\widehat{\mathbb{A}})$, $||\frac{\partial \varphi}{\partial x_i}||_s \leq M$ and $||\frac{\partial^2 \varphi}{\partial x_i \partial x_j}||_s \leq M$ for $i, j = 1, 2, \ldots, m$.

Here $\widehat{\mathbb{A}}$ is primitively recursively Archimedian subfield of $\mathbb{R}$, with PR splitting

T h e o r e m. Let $M, p \geq 2$ be integers and $A, B_1, \ldots, B_m \in M_n(\mathbb{R}_p)$ be fixed matrices satisfying the conditions in (1). Then the solution operator $\varphi \mapsto \mathbf{u}$ for (1) is a PR-computable function (uniformly on m, n) from $C_s^{p+1}(Q, \mathbb{R}^n)$ to $C^p_{sL_2}(H, \mathbb{R}^n)$, with the same constraints on φ as in the previous theorem.

III. Complexity classification of differential equations, in comparison to other classifications

General PDEs: $\begin{cases} Lu(x) = f(x), x \in \Omega \subset \mathbb{R}^k \\ \mathcal{L}u(x)|_\Gamma = \varphi(x\,|_\Gamma), \Gamma \subseteq \partial\Omega. \end{cases}$

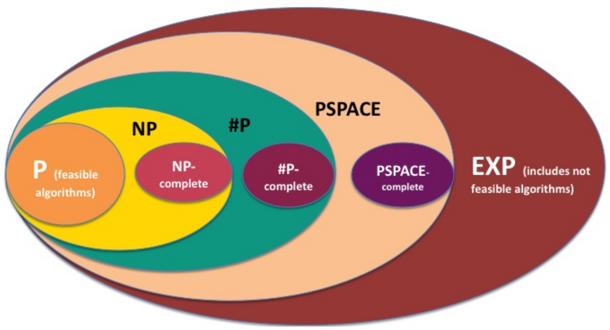
Differential operator:

$$Lu = \sum_{|\alpha|=k} a_\alpha(D^{k-1}\mathbf{u}, \ldots, \mathbf{u}, x) D^\alpha \mathbf{u} + a_0(D^{k-1}\mathbf{u}, \ldots, \mathbf{u}, x), \quad D^\alpha \mathbf{u} = \frac{\partial^{\alpha_1} \ldots \partial^{\alpha_k}}{\partial x_1^{\alpha_1} \ldots \partial x_k^{\alpha_k}}$$

Main **types of** L:

- Linear: constant or variable coefficients; quasilinear; nonlinear
- Hyperbolic, parabolic, elliptic, subelliptic, subparabolic, ...;
- Initial functions: analytic, C^k-smooth, Sobolev functions

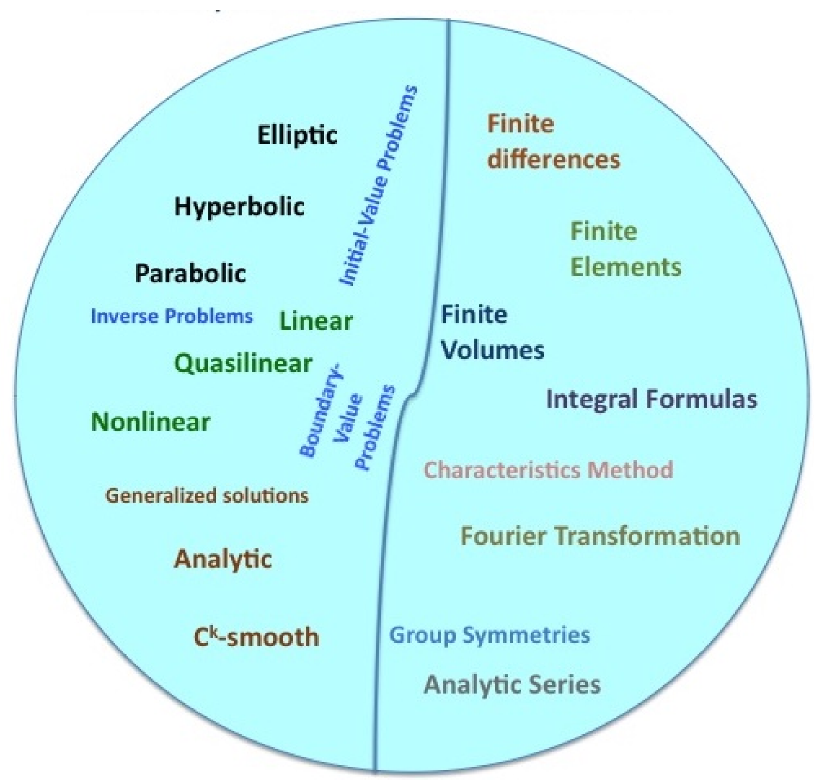
▷ **Complexity hierarchy:**



$$L\Big(t, \vec{x}, \vec{\mathbf{u}}, \frac{\partial \vec{\mathbf{u}}}{\partial t}, \frac{\partial \vec{\mathbf{u}}}{\partial x_i}, \ldots\Big) = 0$$

Initial functions; Coeff. of L $\mapsto$ $\vec{\mathbf{u}}$ **?**

▷ **The variety of PDEs and methods to solve them:**



Main **types of** L and **classification achievements**:

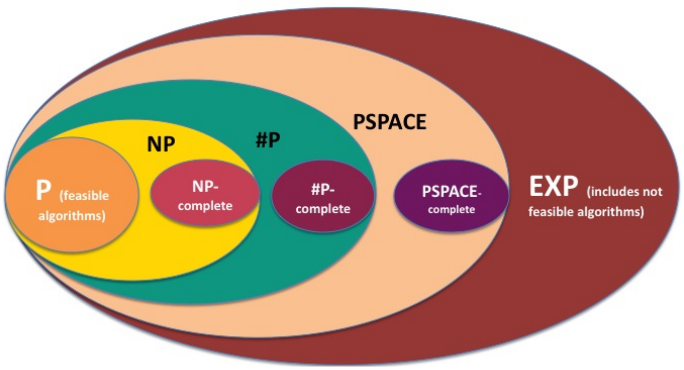
- Linear: constant (♯**P**) or variable (**PSPACE**) coefficients; quasilinear (**2-EXP** upper bound); nonlinear

- Hyperbolic, parabolic, elliptic, subelliptic, subparabolic, ...; (**all in the same complexity class depending on linear/nonlinear, constant/variable coefficients)**

- Initial functions: analytic(**P for linear constant and variable coefficients**, C^k-smooth, Sobolev functions

(j.w.w. Ivan Koswara, Gleb Pogudin and Martin Ziegler, 2019-2022)

Real-valued analog of the Complexity hierarchy



$$\mathbb{R}NC^i \subset \mathbb{R}PTIMESPACE(\log^i n) \subseteq$$

$$\subseteq \mathbb{R}NC^{2i} \ldots \subseteq \mathbb{R}PTIME \subseteq \mathbb{R}\#P_1 \subset$$

$$\subseteq \mathbb{R}\#P \subseteq \mathbb{R}\#P^{\#P} \subseteq$$

$$\subseteq \mathbb{R}PSPACE = \mathbb{R}PAR \subseteq \mathbb{R}EXP$$

- K.-I. Ko, H. Friedman. Computational Complexity of Real Functions, 1982.
- K.-I. Ko. Complexity Theory of Real Functions, 1991.
- A. Kawamura, S. A. Cook. Complexity Theory for Operators in Analysis, 2010.
- A. Kawamura, M. Ziegler. Invitation to Real Complexity Theory: Algorithmic Foundations to Reliable Numerics with Bit-Costs (arXiv), 2018.
- K. Weihrauch. Computable Analysis, 2000.
- M. B. Pour-El, J. I. Richards. Computability in Analysis and Physics (Perspectives in Logic), 2017.

PROGRESS IN THEORETICAL COMPUTER SCIENCE

Complexity Theory of Real Functions

Ker-I Ko

Birkhäuser

REAL BIT-COMPLEXITY

▷ **Main real complexity classes**

◇ For real functions

Def. Computing $f :\subseteq \mathbb{R} \to \mathbb{R}$ in time $t : \mathbb{N} \to \mathbb{N}$ means, on input $a_m \in \mathbb{Z}$ s.th.

$$|x - a_m/2^m| \leq 1/2^m,$$

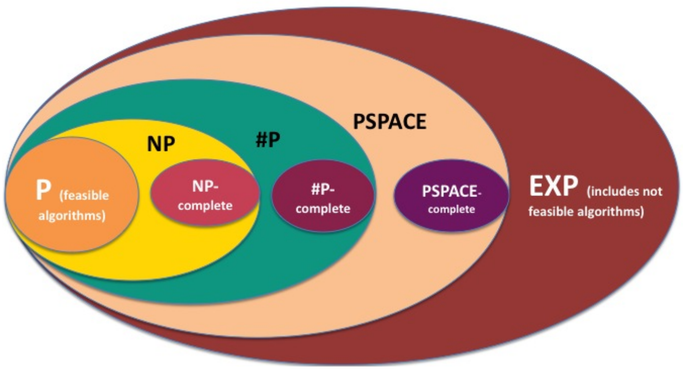
to output $b_n \in \mathbb{Z}$ s.th.

$$|f(x) - b_n/2^n| \leq 1/2^{\mathbf{n}},$$

in $\leq t(\mathbf{n})$ steps.

- $\mathbb{R}$**PTIME** if t(n)=poly(n)
- $\mathbb{R}$**EXP** if t(n)=exp(n)
- $\mathbb{R}$**PSPACE**: if the amount of memory s(n) is bounded polynomially in n

Complexity Hierarchy and real-valued problems



◇ Detailed and intuitive proofs for these examples:
https://kaist.theoryofcomputation.asia/20cs700
(by M.Ziegler, Section V)

◇ Survey on ODEs: D.Graca, N.Zhong. Computability of differential equations, 2021.

- $\max f$: **NP**-complete; **P** for analytic f
- $\int_0^x f(t)dt$: $\sharp$**P**-complete; **P** for analytic f
- $\int_0^1 f(t)dt$: $\sharp$**P**$_1$-complete; **P** for analytic f
- Solutions of ODEs $[\frac{d\vec{u}}{dt} = f(t, \vec{u}), \quad \vec{u}(0) = \vec{u}_0]$: **PSPACE**-complete in general (Kawamura 2010; Kawamura, Ota, Rösnick, Ziegler 2014)
  P for analytic f (Müller, Moiske 1993; Bournez, Graça, Pouly 2011; Pouly, Graça 2016; Kawamura, Steinberg, Thies 2018)

Exact Real Computation

◇ Approach:

- Exact Real Computation has $\mathbb{R}$ (real numbers) as **exact** data type

▷ Computations **approximate** output to **guaranteed** precision 2^{-n} given by the user (i.e., computes **any** n digits versus fixed 53 in double precision)

- Computing a function $t \mapsto \mathbf{u}(t)$:

$$|t - \frac{t_m}{2^m}| < 2^{-m} \to ||\mathbf{u}(t) - \frac{u_n}{2^n}|| < 2^{-\mathbf{n}}$$

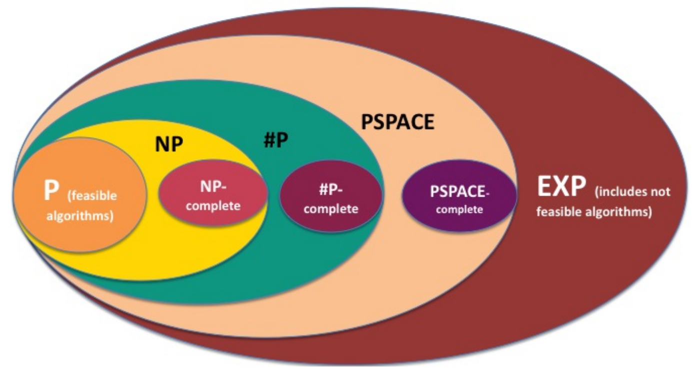
t_m, u_n integers, $m = m(n)$ modulus of continuity of $\mathbf{u}$

- Exact Real Computation packages: iRRAM, ARIADNE, Aern

- We aim to
  - Develop the necessary theory (complexity classification!)
  - Create Exact Real Computation solvers for PDEs, in iRRAM (a library in C++)

Overall goals:

- Provide a bit-cost classification of real-valued (continuous) problems:
  - Studying complexity of particular algorithms and attempts to optimize them: 2^n, n^k, $\log^k n$?
  - Proving optimality results: what is the **best possible** algorithm for the problem?



◇ "discrete" complexity: n=length of the input

◇ "real" complexity: n = number of bits of the output, to obtain the precision 2^{-n}:

$$|t - \frac{t_m}{2^m}| < 2^{-m} \to ||\mathbf{u}(t) - \frac{u_n}{2^n}|| < 2^{-\mathbf{n}}$$

- Provide optimal and reliable algorithms and implement them in Exact Real Computation packages (iRRAM, Ariadne, Aern)

Sewon Park, Franz Brauße, Pieter Collins, SunYoung Kim, Michal Konečný, Gyesik Lee, Norbert Müller, Eike Neumann, Norbert Preining, Martin Ziegler. *Foundation of Computer (Algebra) ANALYSIS Systems: Semantics, Logic, Programming, Verification* (arXiv) 2021

Example: Evolutionary PDEs

Linear; variable coefficients; includes hyperbolic and parabolic

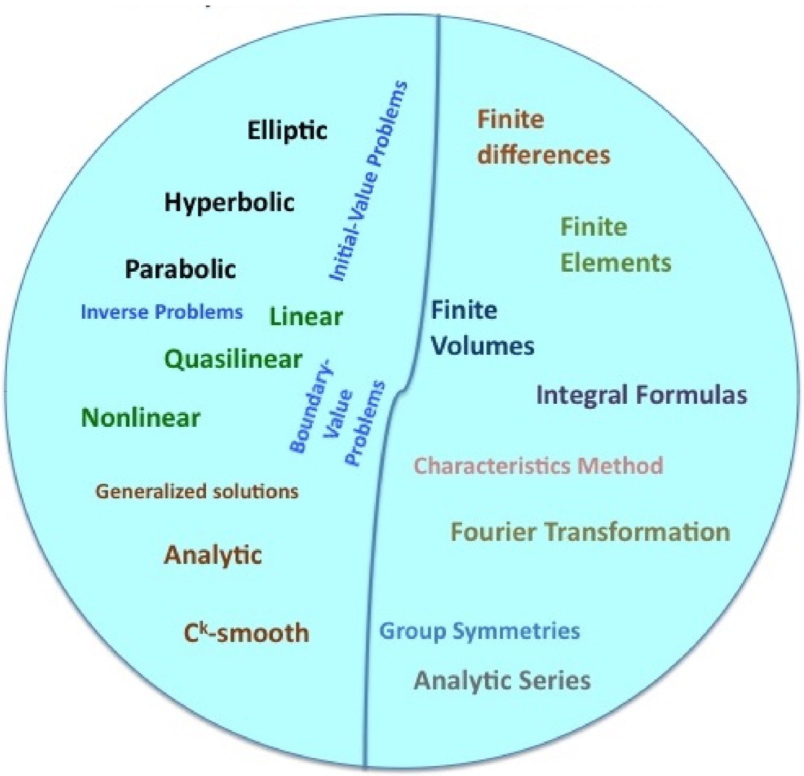
$$\begin{cases} \frac{\partial \vec{u}}{\partial t} = \sum_{|\vec{\jmath}|} B_{\vec{\jmath}}(\vec{x}) \cdot \partial^{\vec{\jmath}} \vec{u}, & t \in [0,1],\ \vec{x} \in [0,1]^d, \\ \vec{u}(0,\vec{x}) = \varphi(\vec{x}), \quad \vec{x} \in [0,1]^d, & \mathcal{L}\vec{u}\,|_{\partial[0,1]^d} = 0, \end{cases}$$

where $\partial^{\vec{\jmath}} = \frac{\partial^{j_1}}{\partial x_1^{j_1}} \cdots \frac{\partial^{j_d}}{\partial x_d^{j_d}}$, $\vec{u} = (u_1, \ldots, u_{d'})$.

Questions: **1.** Given polynomial-time computable matrix coefficient functions $B_{\vec{\jmath}}(\vec{x})$ and initial vector-function $\varphi(\vec{x})$ (from certain functional classes), find to which complexity class does the (unique) solution $\vec{u}(t,\vec{x})$ belong**?**

2. Find the optimal complexity class.

$$L\Big(t, \vec{x}, \vec{\mathbf{u}}, \frac{\partial \vec{\mathbf{u}}}{\partial t}, \frac{\partial \vec{\mathbf{u}}}{\partial x_i}, \ldots\Big) = 0$$



Goals for PDEs

- Develop a uniform framework for solving (important classes of) PDEs with guaranteed arbitrary precision given by the user

$$||\vec{\mathbf{u}} - u^{(n)}|| < 2^{-n}$$

- Classify PDEs by their algorithmic complexity.

- Based on this classification we develop and implement optimal and reliable algorithms. (S., Steinberg, Thies, Ziegler 21; J.w.w.: Semertzidis, Thies 22 in progress, Efendiev, Hoang, Thies 22 in progress)

ODEs	Evolutionary PDEs (including Hyperbolic and Parabolic)	Other Types of PDEs (including Elliptic)
$\begin{cases} \frac{d}{dt}\vec{u} = f(t,\vec{u}), \\ \vec{u}(0) = \vec{v} \end{cases}$ (1) ▷ $f \in$ P **analytic** $\Longrightarrow \vec{u} \in$**P** [Müller, Moiske'93; Bournez, Graça, Pouly'11; Pouly, Graça'16; Kawamura, Steinberg, Thies'18] ▷ $f \in$ P linear $\Longrightarrow \vec{u} \in$ Log^2-SPACE [Koswara, S., Ziegler'19] ▷ $f \in$P Lipschitz or $C^1 \Longrightarrow \vec{u}$ **PSPACE-complete** [Kawamura'10]	$\begin{cases} \vec{u}_t = \sum_{\lvert\vec{j}\rvert} \mathbf{B}_{\vec{j}}(\vec{x}) \cdot \partial^{\vec{j}}\vec{u}, \\ \vec{u}(0,\vec{x}) = \varphi(\vec{x}) \end{cases}$ ▷ $\varphi, B_j \in$P **analytic** $\Longrightarrow \vec{u} \in$**P** ([Koswara, S., Ziegler'19] ; [S., Steinberg, Thies, Ziegler'21] uniform version; [S., Selivanov'21] analysis of dependence on constant matrix coefficients $B_j = B_j^*$ over various real closed fields); ▷ $\varphi, B_j \in$P: C^k, $k \geq 1$ (well posed) $\Longrightarrow$ [Koswara, Pogudin, S., Ziegler] – $\vec{u}$ $\in$**PSPACE-complete** (general case) – $\vec{u} \in$**♯P** (constant periodic case) – $\vec{u}$ **♯P$_1$-'hard'** (heat equation) – $\vec{u} \in$P for constant mutually commuting matrices ▷ for the quasilinear case $B_j = B_j(\vec{x},\vec{u})$ upper bound **2-EXP** [Koswara, S., Ziegler'19].	$\begin{cases} \sum_{j=1}^{d} \frac{\partial^2}{\partial x_j^2} u = f \text{ in } B^d \\ u\mid_{\partial B^d} = g(\vec{x}) \end{cases}$ ▷ $f, g \in$P $\Longrightarrow u \in$**♯P; ♯P$_1$-'hard'** [Kawamura, Steinberg, Ziegler'17]

IV. Conclusion

- Computable Analysis
- Real Complexity Theory
- Exact Real Computation

Help to build a bridge between (1) classical "discrete" computability and complexity theories, and (2) classical "continuous" analysis and numerical/analytical methods of solving differential equations.

Complexity classification allows to design rigorous solution methods within the uniform Exact Real Computation framework.

In particular it applies to the above mentioned Motivational Examples.

Happy Birthday!