

Bit-Complexity of Solving Systems of Linear Evolutionary Partial Differential Equations*

Ivan Koswara¹[0000–0002–9311–6840], Gleb Pogudin²[0000–0002–5731–8242],
Svetlana Selivanova¹[0000–0002–8180–0311], and Martin
Ziegler¹[0000–0001–6734–7875]

¹ School of Computing, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141 Republic
of Korea {chaoticiak,sseliv,ziegler}@kaist.ac.kr

² LIX, CNRS, École Polytechnique, Institut Polytechnique de Paris, 91128 France
gleb.pogudin@polytechnique.edu

Abstract. *Finite Elements* are a common method for solving differential equations via discretization. Under suitable hypotheses, the solution $\mathbf{u} = \mathbf{u}(t, \vec{x})$ of a well-posed initial/boundary-value problem for a linear evolutionary system of PDEs is approximated up to absolute error $1/2^n$ by repeatedly (exponentially often in n) multiplying a matrix $\mathbf{A}_n$ to the vector from the previous time step, starting with the initial condition $\mathbf{u}(0)$, approximated by the spatial grid vector $\mathbf{u}(0)_n$. The dimension of the matrix A_n is exponential in n , which is the number of the bits of the output.

We investigate the bit-cost of computing exponential powers and inner products $\mathbf{A}_n^K \cdot \mathbf{u}(0)_n$, $K \sim 2^{\mathcal{O}(n)}$, of matrices and vectors of exponential dimension for various classes of such *difference schemes* $\mathbf{A}_n$. Non-uniformly fixing any polynomial-time computable initial condition and focusing on single but arbitrary entries (instead of the entire vector/matrix) allows to improve naïve exponential sequential runtime EXP: Closer inspection shows that, given any time $0 \leq t \leq 1$ and space $\vec{x} \in [0; 1]^d$, the computational cost of evaluating the solution $\mathbf{u}(t, \vec{x})$ corresponds to the discrete class PSPACE.

Many partial differential equations, including the Heat Equation, admit difference schemes that are (tensor products of constantly many) circulant matrices of constant bandwidth; and for these we show exponential matrix powering, and PDE solution computable in #P. This is achieved by calculating individual coefficients of the matrix’ multivariate companion polynomial’s powers using Cauchy’s Differentiation Theorem; and shown optimal for the Heat Equation. Exponentially powering twoband circulant matrices is established even feasible in P; and under additional

*Supported by the National Research Foundation of Korea (grant 2017R1E1A1A03071032) and by the International Research & Development Program of the Korean Ministry of Science and ICT (grant 2016K1A3A7A03950702) and by the NRF Brain Pool program (grant 2019H1D3A2A02102240). GP was supported by NSF grants CCF-1564132, CCF-1563942, DMS-1853482, DMS-1853650, and DMS-1760448, by PSC-CUNY grants #69827-0047 and #60098-0048. We thank Lina Bondar’ for a helpful discussion on different versions of the Sobolev Embedding Theorem (Example 2b)).

conditions, also the solution to certain linear PDEs becomes computable in P.

Keywords: Reliable Computing · Bit-Cost · Partial Differential Equations.

1 Introduction and Summary of Contributions

Computable Analysis [31] provides a framework for rigorous computability and complexity investigations of computational problems over real numbers and functions by approximation up to guaranteed absolute error $1/2^n$ [15, 32, 2, 10]. This has been applied to ordinary [11, 14] and partial [27, 28, 30] differential equations. It allows to prove asymptotic optimality of numerical algorithms by relating the intrinsic computational bit-cost of a problem to a classical discrete complexity class [8, 13].

The present work considers general classes of systems of linear evolutionary partial differential equations (PDEs).

In order to solve some system of ordinary differential equations (ODEs) $\partial_t \mathbf{u} = \tilde{f}(\mathbf{u}, t)$, common numerical approaches—such as Euler’s Method and its refinements—discretize time $t \in [0; 1]$ into steps $\tau \ll 1$: From the fixed initial value $\mathbf{u}(0) = \mathbf{u}_0$ at $t = 0$ they iteratively proceed to approximations $\mathbf{u}(\tau)$, $\mathbf{u}(2\tau)$, $\dots \mathbf{u}(M \cdot \tau)$. In order for the last one to approximate $\mathbf{u}(1)$ up to error $1/2^n$, the number $M = 1/\tau \in \mathbb{N}$ of steps is generally exponential in (the number of bits of the output) n ; and the problem thus seen to belong to the discrete complexity class EXP[†]. Closer inspection improves that to PSPACE [15, §7.2], which has been proven best possible in general [8]. Bit complexity is measured w.r.t. the output precision parameter n .

Evolutionary PDEs generalize ODEs: by replacing the right-hand side function $\tilde{f} = \tilde{f}(\mathbf{u}, t)$ with an operator $\mathcal{A}$, commonly involving spatial derivatives. Solutions $\mathbf{u}(t)$ accordingly now take values in some function space, rather than in Euclidean. The mathematical theory of PDEs is considerably more involved than that of ODEs [3] regarding existence, uniqueness, and continuous dependence of solutions (=well-posedness in the sense of Hadamard): Recall that one of the Millennium Prize Problem asks such questions for Navier-Stokes’ Equation. Computability investigations of PDEs have challenged the Church-Turing Hypothesis [25, 33, 30]. The present work considers linear evolutionary PDEs with initial and boundary conditions:

$$\begin{cases} \mathbf{u}_t = \mathcal{A}\mathbf{u}, & 0 \leq t \leq 1, \quad \vec{x} \in \Omega, \\ \mathbf{u}|_{t=0} = \varphi(\vec{x}), & \vec{x} \in \Omega, \\ \mathcal{L}\mathbf{u}(t, \vec{x})|_{\partial\Omega} = 0, & (t, \vec{x}) \in [0, 1] \times \partial\Omega \end{cases} \quad (1)$$

where $\Omega = [0, 1]^d$ is the unit cube (for technical simplicity); $\partial\Omega$ is its boundary; the solution $\mathbf{u} = (u_1, \dots, u_e) = \mathbf{u}(t, \vec{x})$ is an unknown vector function on Ω ;

[†]Definitions of the real-valued counterparts of the complexity classes are given in Subsection 1.3; for simplicity we use same notation as for the “discrete” case for them.

$\mathcal{L}$ in the boundary condition is a linear differential operator of order less than the order of the differential operator $\mathcal{A}$. The coefficients of $\mathcal{A} = \sum_{|\vec{j}|} \mathbf{B}_{\vec{j}}(\vec{x}) \cdot \partial^{\vec{j}}$ are $e \times e$ matrices $\mathbf{B}_{\vec{j}}$ that may depend on $\vec{x}$, but not on t (autonomous case), $\vec{j} = (j_1, \dots, j_d)$ denotes a multi-index ($|\vec{j}| = j_1 + j_2 + \dots + j_d$), $\partial^{\vec{j}} = \partial_1^{j_1} \dots \partial_d^{j_d}$ the induced differential operator (with $\partial_k^{j_k} = \frac{\partial^{j_k}}{\partial x_k^{j_k}}$), and $\varphi(\vec{x})$ is the initial condition. Note that the equations (1) are linear in the derivatives, but the matrix coefficients $\mathbf{B}_{\vec{j}}$ can depend on $\vec{x}$ nonlinearly.

Example 1. An important and rich class of PDEs of form (1) are the (first-order) symmetric hyperbolic systems

$$\mathbf{B}_0(\vec{x}) \cdot \mathbf{u}_t = \sum_{j=1}^d \mathbf{B}_j(\vec{x}) \cdot \partial_{x_j} \mathbf{u} \quad (2)$$

with $\mathbf{B}_j(\vec{x}) = \mathbf{B}_j^*(\vec{x})$, $j = 0, 1, \dots, d$; $B_0(\vec{x}) > 0$. The corresponding differential operator is $\mathcal{A} = \sum_{j=1}^d \mathbf{B}_0^{-1}(\vec{x}) \cdot \mathbf{B}_j(\vec{x}) \cdot \partial_j$. This class includes the linear acoustics, elasticity and Maxwell equations [27]. Also the (second-order) Wave Equation $u_{tt} = \Delta u$ ($= \sum_{j=1}^n \frac{\partial^2 u}{\partial x_j^2}$) and many others can be reduced to such a system by introducing extra unknown functions.

The Heat Equation $u_t = \Delta u$ is not of the form (2), but still of the form (1). Periodic boundary conditions on the unit cube are captured by

$$\begin{aligned} \mathcal{L}\mathbf{u}(t, x_1, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_d) &:= \mathbf{u}(t, x_1, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_d) - \\ &\quad \mathbf{u}(t, x_1, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_d) \end{aligned} \quad (3)$$

for first-order systems, and include similar conditions on spacial derivatives up to $l - 1$ order for l -order systems.

Under suitable hypotheses, Euler's method generalizes from ODEs to evolutionary PDEs (1): by discretizing now both physical time and space, the latter with some grid of sufficiently (=exponentially) small width $h \ll 1$. This turns the initial condition φ into a vector of exponentially large dimension $\mathcal{O}(1/h)$. The right-hand side linear operator $\mathcal{A}$ may be approximated by a matrix $\mathbf{A}$, often referred to as *difference scheme*, see Definition 1. And if the evolution equation is autonomous, said matrix does not depend on time. In this case repeated time-stepping $\mathbf{u}(t) \mapsto \mathbf{u}(t + \tau) = \mathbf{A} \cdot \mathbf{u}(t)$ amounts to repeated (M -fold) multiplication by $\mathbf{A}$, i.e., to (exponential) matrix powering $\mathbf{A}^M$.

Now all three, the discretized initial condition $\mathbf{u}(0) = \varphi$ and the matrix $\mathbf{A}$ and the resulting approximation to $\mathbf{u}(1)$, have dimension $K = \mathcal{O}(1/h)$ exponential in n : leaving no chance for sub-exponential computational cost. More relevant is therefore the following question:

Question 1. Fix polynomial-time computable initial condition, fix polynomial-time computable matrix coefficients $\mathbf{B}_{\vec{j}}$ in the right-hand side of PDE (1), and similarly for boundary condition $\mathcal{L}$. Now consider only $(t, \vec{x}) \in [0; 1] \times \Omega$ as input: What is the bit-cost (measured w.r.t. the parameter n) of approximating the solution $\mathbf{u}(t, \vec{x})$ at time t and point $\vec{x}$ up to absolute error $1/2^n$?

Thus, non-uniformly fixing all data of exponential ‘size’ (formally: from spaces of exponential entropy [16,32,12]) and restricting to polynomial ‘size’ inputs $(t, \vec{x}) \in [0; 1] \times \Omega$ avoids information-theoretic exponential lower complexity bounds.

Compact domains ensure that one can restrict to complexity considerations in terms of one parameter n [31, Theorem 7.2.7] and does not need to resort to second-order complexity [9]. We use n as parameter for producing approximations up to error bound $1/2^n$, not $1/n$, following the conventions of Real Complexity Theory [15, Definition 2.7]; see also Remark 1 below.

Note that PSPACE has been proven best possible for solving a certain *non-linear* ODE [8]. Poisson’s elliptic (i.e. non-evolutionary) PDE has been established to similarly characterize $\#P_1$ [13]. Computation on grids of size $\mathcal{O}(N)$ have been shown computable in $\mathcal{O}(\log N)$ parallel runtime [22]. In our terminology of grid width $h \sim 1/2^n$ for guaranteed output approximation error $1/2^n$, this means $N = \mathcal{O}(1/h^d)$ and parallel runtime $\mathcal{O}(nd)$. That would amount to complexity class $\text{PAR} = \text{PSPACE}$, were it not for the superpolynomial number $\mathcal{O}(N/\log N) = \mathcal{O}(2^{nd}/nd)$ of processors. Our previous work [17] has rigorously and in the sense of Question 1 established solutions to a large class of linear first-order evolutionary PDEs computable in PSPACE, and under additional hypotheses even in $\#P^{\#P}$.

The main result of the present paper, Theorem 2, improves the latter to $\#P$. It applies, in particular, to the Heat Equation, where we show $\#P_1$ as optimal.

1.1 Main Result and Overview

The complexity considerations in this work refer to (real counterparts, formalized in Subsection 1.3, of) the classical hierarchy[§] commonly conjectured proper:

$$\text{NC} \subseteq \text{P} \subseteq \text{NP} \subseteq \#P \subseteq \#P^{\#P} \subseteq \dots \subseteq \text{PSPACE} = \text{PAR} \subseteq \text{EXP} . \quad (4)$$

The following hypotheses are very natural and hold for many PDEs including many of the ones mentioned in Example 1, see Example 2 for more detail. For the notation of (iii) see Subsection 1.2; $\|\varphi\|_{C^l(\bar{\Omega})} = \sup_{\vec{x} \in \bar{\Omega}} \sum_{|\vec{j}| \leq l} |\partial^{\vec{j}} \varphi(\vec{x})|$.

Hypotheses 1 (i) *The problem (1) is well-posed (Hadamard) in that the classical solution $\mathbf{u}(t, \vec{x})$ to (1) exists, is unique and depends continuously on the initial data in the following sense:*

$$\varphi(\vec{x}) \in \mathcal{C}^l(\bar{\Omega}), \quad \mathbf{u}(t, \vec{x}) \in \mathcal{C}^2([0, 1] \times \bar{\Omega}), \quad \|\mathbf{u}\|_{\mathcal{C}^2([0, 1] \times \bar{\Omega})} \leq C_0 \|\varphi\|_{C^l(\bar{\Omega})}, \quad (5)$$

for some fixed $C_0, l \geq 2$.

(ii) *The initial functions $\varphi(\vec{x})$ and matrix coefficients $\mathbf{B}_j(\vec{x})$ as well as their partial derivatives up to order l are computable in P .*

[§]The reader may forgive us for identifying decision and function complexity classes.

- (iii) The system (1) admits a difference scheme $\mathbf{A}_{h(n)}$ (see (6) below) which is computable in P , and its solution $u^{(n)}$ converges to the solution $\mathbf{u}$ of (1) w.r.t. the maximum norm on the uniform grid $G_{h(n)}$ with the step $h = h(n)$:

$$\max_{x \in G_{h(n)}} |\mathbf{u}|_{G_{h(n)}} - u^{(n)}| < C \cdot h(n), \quad C \text{ does not depend on } n.$$

Note that technically a difference scheme is a family $\mathbf{A}_{h(n)}$ of matrices of dimension growing exponentially in $n \rightarrow \infty$ such as to approximate the operator $\mathcal{A}$ with increasing precision; the approximating solution $u^{(n)}$ is a sequence of vectors of dimension growing exponentially in n . See Definitions 5, 7 of Subsection 1.3 for adjustment of the complexity classes to this case.

The main results of the present paper are collected in the following

- Theorem 2.** a) The solution $\mathbf{u}$ of (1) under Hypotheses 1 is computable in PSPACE .
 b) If additionally the difference scheme $\mathbf{A}_h$ from (iii) is a sum of tensor products of circulant block matrices of constant bandwidth (as formalized in the hypothesis of Theorem 4), then evaluating the solution $(t, \vec{x}) \mapsto \mathbf{u}(t, \vec{x})$ of (1) is computable in $\#\mathsf{P}$.
 c) Evaluating the solution $\mathbf{u}$ of (2) is computable in P if the matrices $\mathbf{B}_j$ are constant and mutually commute for $j = 0, 1, \dots, d$.
 d) For the Heat Equation $u_t = \Delta u$ there exists a polynomial time computable initial condition φ such that the solution u is classical but cannot be computed in polynomial time unless $\mathsf{P}_1 = \#\mathsf{P}_1$.

Item b) harnesses a particular structure common to difference schemes, formalized in Theorem 4 below. Intuitively, in 1D the locality of the grid discretization of the differential operator yields a difference scheme $\mathbf{A}_{h(n)}$ with constant bandwidth, and periodic boundary conditions yield to a circulant structure. Higher-dimensional Euclidean domains translate to tensor products of such (families of) matrices.

Example 2. Many evolutionary linear PDEs admit difference schemes that satisfy the hypotheses of Theorem 2 b), and thus can be computed in $\#\mathsf{P}$, including:

- a) the Heat Equation with periodic boundary conditions and polynomial time computable initial function: see [19, §2.11] for maximum norm difference scheme convergence, and e.g. [3] for well-posedness.
 b) the Wave Equation with periodic boundary conditions and polynomial time computable initial functions. Indeed, the wave equation admits a max-norm convergent difference scheme under additional smoothness assumptions, see Theorem 3.1 of [20] for the two-dimensional case, given that $u(t, x, y) \in C^{(4,5)}([0, T] \times \bar{\Omega})$. The continuous dependence condition (5) can be verified combining the well-known continuous dependence w.r.t. L_2 -norms, and Sobolev Embedding Theorem, see e.g. [3, §5.6.3].

c) Note that the two-dimensional acoustics system

$$\begin{cases} \rho_0 \frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = 0, \\ \rho_0 \frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} = 0, \\ \frac{\partial p}{\partial t} + \rho_0 c_0^2 \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \end{cases}$$

can be equivalently reduced to the two-dimensional wave equation (e.g. [4]); see also [5] for other examples of symmetric hyperbolic systems (2) (with constant coefficients $\mathbf{B}_j$), which are equivalent to higher-order wave equations.

Note that without assuming the boundary conditions periodic, the solution u in the examples a), b), c) above can be computed in PSPACE, according to Theorem 2 a).

Subsection 1.2 collects notational conventions and recalls some basic definitions. Subsection 1.3 formalizes computational bit-complexity theory of real vectors and matrices of exponential dimension. Section 2 presents the proof (sketches) to our main Theorem 2.

1.2 Notation

We use $n \in \mathbb{N}$ to parametrize the absolute output approximation error bound $1/2^n$; $d \in \mathbb{N}$ is the dimension of the torus $\Omega = [0; 1]^d \bmod 1$ as compact spatial domain of the partial differential equation under consideration and $e \in \mathbb{N}$ denotes the dimension of the solution function vector $\mathbf{u}$.

Definition 1 (Difference schemes).

a) Consider, for any positive integer N , the uniform rectangular grid G_N on Ω defined by the points

$$\left(\frac{i_1 - \frac{1}{2}}{2^N}, \frac{i_2 - \frac{1}{2}}{2^N}, \dots, \frac{i_m - \frac{1}{2}}{2^N} \right)$$

where $1 \leq i_1, i_2, \dots, i_m \leq 2^N$. Let $h = 1/2^N$ be the corresponding spatial grid step and τ be a time step. Denote $G_N^\tau = G_N \times \{l\tau\}_{l=1}^M$, where M is the number of time steps. The choice of steps h and τ , depending on the output precision parameter n , is specified below in Subsection 2.3. We consider the following grid norm: $|g^{(h)}| = \max_{x \in G_N} |g^{(h)}(x)|$.

b) For a linear differential operator $\mathcal{A}$, the matrix $\mathbf{A}_{(h)}$ (with the grid step $h = h(n)$) defines the corresponding difference scheme

$$u^{(h, (l+1)\tau)} = \mathbf{A}_{\mathbf{h}} u^{(h, l\tau)}, \quad u^{(h, 0)} = \varphi^{(h)} \quad (6)$$

- under consideration. Its entries are denoted $(\mathbf{A}_{(h)})_{I,J}$, $1 \leq I, J \leq K$. Here $K \sim 2^{\mathcal{O}(n)}$ is the dimension of the vectors $\mathbf{u}^{(h,m\tau)}$ approximating the solution $\mathbf{u}(m\tau, \vec{x},)$ at time $m\tau \leq 1$, i.e., for $1 \leq m \leq M := 1/\tau \sim 2^n$. $\tau, h \sim 1/2^n$ denote the temporal and spatial grid widths, respectively. Generally speaking, capital letters denote quantities (ranging up to) exponential in n .
- c) The solution $u^{(h)}$ of the difference scheme (6) converges to the solution u of (1) if there is a constant C not depending on h and τ such that

$$|u|_{G_h^\tau} - u^{(h)}| \leq Ch^p . \quad (7)$$

Due to the Lax Convergence Theorem, a difference scheme (6) converges in the sense of (7) to the solution of the PDEs (1) if and only if it is *approximating* and *stable*; see e.g. [29]. The latter means that the matrix A_h has bounded powers, see Definition 2 b) below.

Definition 2 (Matrices).

- a) Equip vectors with the maximum norm, and matrices with the induced operator norm: $\|\mathbf{u}\| = \max_j |u_j|$, $\|\mathbf{A}\| = \max \{\|\mathbf{A} \cdot \mathbf{u}\| / \|\mathbf{u}\|\}$.
- b) A square matrix $\mathbf{A}$ with entries $\mathbf{A}_{I,J}$ ($0 \leq I, J < K$) is said to have bounded powers if its powers are uniformly bounded, i.e., iff there exists some $C \in \mathbb{N}$ such that $\|\mathbf{A}^M\| \leq C$ holds for all $M \in \mathbb{N}$.
Similarly for a family $\mathbf{A}_n$ of square matrices of possibly varying format, $n \in \mathbb{N}$: Here $\|\mathbf{A}_n^M\|$ must be bounded independently of both $M \in \mathbb{N}$ and of $n \in \mathbb{N}$.
- c) For a (not necessarily commutative) ring $\mathcal{R}$, let $\mathcal{R}_D^{N \times N}$ denote the vector space of $N \times N$ matrices of bandwidth $< D$:

$$\mathbf{A} \in \mathcal{R}_D^{N \times N} \quad \Leftrightarrow \quad |I - J| \geq D \Rightarrow \mathbf{A}_{I,J} = 0.$$

The (one-)norm of an *integer* multi-index $\vec{j} = (j_1, \dots, j_L) \in \mathbb{Z}^L$ is $|\vec{j}| = j_1 + \dots + j_L$. We write “ $\vec{j} \geq \vec{0}$ ” to indicate non-negative (i.e. natural number) multi-indices. $D = \max \{|\vec{j}| : \mathbf{B}_{\vec{j}} \neq 0\}$ denotes the *order* of the PDE (1).

A main tool in our algorithms translates circulant matrix powering to polynomial powering. Indeed, the above notions have immediate counterparts:

Definition 3 (Polynomials). Fix a (not necessarily commutative) ring $\mathcal{R}$.

- a) A polynomial in L commuting variables of componentwise degree less than $\vec{D} \in \mathbb{N}^L$ has the form

$$P(\vec{X}) = P(X_1, \dots, X_L) = \sum_{\vec{0} < \vec{j} \leq \vec{D}} p_{\vec{j}} \vec{X}^{\vec{j}} = \sum_{\vec{j} \geq \vec{0}, |\vec{j}| < \vec{D}} p_{\vec{j}} \prod_{\ell=1}^L X_{\ell}^{j_{\ell}} .$$

Write $\mathcal{R}_{\vec{D}}[X_1, \dots, X_L] = \mathcal{R}_{\vec{D}}[\vec{X}]$ for the vector space of such polynomials.

- b) An L -variate Laurent polynomial of componentwise degree $< \vec{D}$ has the form

$$P(\vec{X}) = P(X_1, \dots, X_L) = \sum_{-\vec{D} < \vec{j} < \vec{D}} p_{\vec{j}} \vec{X}^{\vec{j}} ,$$

that is, including negative powers of the variables. Write $\mathcal{R}_{\vec{D}}[X_1, X_1^{-1}, \dots, X_L, X_L^{-1}] = \mathcal{R}_{\vec{D}}[\vec{X}, \vec{X}^{-1}]$ for the vector space of such Laurent polynomials; and $P[\vec{X}^{\vec{j}}] := p_{\vec{j}}$ for the coefficient to $\vec{X}^{\vec{j}}$ in P , $\vec{j} \in \mathbb{Z}^L$.

- c) Suppose $\mathcal{R}$ is equipped with a norm $|\cdot|$. Consider the ring $\mathcal{R}[\vec{X}, \vec{X}^{-1}]$ of all (Laurent) polynomials, equipped with the induced norm $\|P\| := \sum_{|\vec{j}|} |p_{\vec{j}}|$. A (Laurent) polynomial P has bounded powers iff $\|P^M\|$ is bounded independently of $M \in \mathbb{N}$.

Similarly for a family P_n of polynomials of possibly varying number of variables, $n \in \mathbb{N}$: Here $\|P_n^M\|$ must be bounded independently of both $M \in \mathbb{N}$ and of $n \in \mathbb{N}$.

Multivariate degree is understood componentwise and w.r.t. strict inequality $<$ (not $=$ nor $\leq$). For example $X^2 \cdot Y^3 + X^3 \cdot Y^2$ has componentwise degree $< \vec{D} = (4, 4)$, but not $< (3, 4)$ nor $< (4, 3)$. An L -variate Laurent polynomial P of componentwise degree $< \vec{D}$ can be converted to an ordinary polynomial by multiplying P with $X_1^{D_1} \dots X_L^{D_L}$.

In the sequel, *exponential* growth is to be understood as bounded by $2^{p(k)}$, $k \rightarrow \infty$, for some polynomial p .

1.3 Real Complexity Theory

In [15], major classical complexity classes have been adapted from the discrete case to the setting of real numbers and (continuous) real functions. There the integer parameter n governing the output approximation error $1/2^n$ replaces the role of the binary input length. Let us recall the definitions of polynomial/exponential time/space computability of real numbers, (fixed-dimensional) real vectors, sequences of real numbers, and partial real functions [15].

Definition 4. a) Computing a **real number** $r \in \mathbb{R}$ means to output, given $n \in \mathbb{N}$, some numerators $a_n \in \mathbb{Z}$ in binary with $|r - a_n/2^{p(n)}| \leq 1/2^n$ for some polynomial $p \in \mathbb{N}[N]$. Such a computation runs in polynomial time (P) if said a_n is output within a number of steps bounded by a polynomial in n . It runs in exponential time (EXP) if the number of steps is bounded exponentially in n . The computation runs in polynomial space (PSPACE) if the amount of memory is bounded polynomially in n .

- b) Computing a **(finite-dimensional) real vector** (in P, EXP, PSPACE) means to compute each of its entries separately (in P, EXP, PSPACE).

- c) Computing a **sequence** $\vec{r} = (r_k) \in \mathbb{R}$ of **real numbers** means to output, given n and k , some $a_{n,k} \in \mathbb{Z}$ in binary with $|r_k - a_{n,k}/2^{p(n+k)}| \leq 1/2^n$ for some polynomial $p \in \mathbb{N}[N]$. Such a computation runs in polynomial time (P) if said $a_{n,k}$ is output within a number of steps bounded by a polynomial in $n+k$. Similarly for exponential time (EXP) and polynomial space (PSPACE).

- d) Computing a **partial real function** $f : \subseteq \mathbb{R} \rightarrow \mathbb{R}$ (w.r.t. some polynomials $q, p \in \mathbb{N}[N]$) means, given $n \in \mathbb{N}$ and any numerator $a \in \mathbb{Z}$ with $|x - a/2^{q(p(n))}| \leq 1/2^{p(n)}$ in binary for some $x \in \text{dom}(f)$, to output some $b =$

$b_n(a) \in \mathbb{Z}$ in binary with $|f(x) - b_n/2^{p(n)}| \leq 1/2^n$. The computation may behave arbitrarily on inputs a that do not satisfy the hypothesis.

Remark 1. In addition to following the conventions of Real Complexity Theory [15, Definition 2.7], we prefer error bound $1/2^n$ (as opposed to $1/n$) for four reasons:

- a) It corresponds to measuring computational cost of discrete problems, such as of integer factorization, in dependence of the binary (as opposed to unary) length n .
- b) It reflects that (for instance Chudnovsky's or Borwein's) algorithms can approximate π in time polynomial in n up to error $1/2^n$ (while error bound $1/n$ is trivial to achieve).
- c) It gives rise to the aforementioned and subsequent and many more [15] numerical characterizations of discrete complexity classes.
- d) The first-order theory of the two-sorted structure $(\mathbb{Z}, 0, 1, +, >) \cup (\mathbb{R}, 0, 1, +, \times, >)$ with 'error embedding' $\iota : \mathbb{Z} \ni n \mapsto 2^{-n} \in \mathbb{R}$ (capturing *Exact Real Computation*) is decidable, while that with $\mathbb{N}_+ \ni n \mapsto 1/n \in \mathbb{R}$ is not [24, Theorem 4.4].

For real vectors of exponential (in some integer parameter k) dimension D_k , our notions of (time/space) complexity are more subtle and we adapt them to the computation of any desired entry rather than of the entire matrix. The complexity of computing said dimension D_k itself must be taken into account as well (see details in the Appendix). Defining real counterparts to $\#P$ is subtle and discussed in detail, since most of our major results refer to it.

When speaking of complexity of computation for **matrices**, we identify a $D \times E$ -dimensional matrix $B = (b_{I,J})$ with the $D \times E$ -dimensional vector $B_{\langle I,J \rangle}$ for the pairing function $\langle I, J \rangle = J + (I + J) \cdot (I + J + 1)/2$.

Definition 5. a) Computing a **sequence** $\vec{r}_k = (r_{k,J})_{J \leq D_k} \in \mathbb{R}^{D_k}$ of **D_k -dimensional real vectors** means to output, given $n, k \in \mathbb{N}$ (in unary) and $J \leq \dim(\vec{r}_k) = D_k$ in binary, some $a_{n,k,J} \in \mathbb{Z}$ in binary with $|r_{k,J} - a_{n,k,J}/2^{p(n+k)}| \leq 1/2^n$ for some polynomial $p \in \mathbb{N}[N]$.

Such a computation runs in polynomial time (P) if said $a_{n,k,J}$ is output within a number of steps bounded by a polynomial in $n + k$ but independently of J . Similarly for exponential time (EXP) and polynomial space (PSPACE).

b) More generally, computing a **sequence** $\vec{f}_k : \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^{D_k}$ of **partial vector functions** (w.r.t. some polynomials $q, p \in \mathbb{N}[N]$) means, given $n, k, J \in \mathbb{N}$ and any numerator $\vec{a} \in \mathbb{Z}^d$ with $|\vec{x} - \vec{a}/2^{q(p(n+k)+k)}| \leq 1/2^{p(n+k)}$ for some $\vec{x} \in \text{dom}(f_k)$ and $J \leq D_k = \dim f_k$, to output some $b = b_{n,k,J}(a) \in \mathbb{Z}$ with $|f_{k,J}(\vec{x}) - b/2^{p(n+k)}| \leq 1/2^n$. Here $\|\vec{y}\| = \max\{|y_1|, \dots, |y_d|\}$ denotes the maximum norm.

Such a computation runs in polynomial time (P) if said $b = b_{n,k,J}(a)$ is output within a number of steps bounded by a polynomial in $n + k$ but independently of J and a . Similarly for exponential time (EXP) and polynomial space (PSPACE).

- c) A sequence (D_k) of natural numbers is computable in unary polynomial time (Pone) if the mapping $0^k \mapsto \text{bin}(D_k) \in \{0, 1\}^*$ is computable in time polynomial in the input length.

Note that a polynomial-time computable real sequence r_k according to (4c) can grow at most exponentially (in k); similarly for both the entries and the dimension D_k of a polynomial-time computable vector sequence according to (4d), and for vector functions according to (5b): in agreement with a sequence D_k computable in unary polynomial time according to (5c) growing at most exponentially in k . Memory-bounded computation here is understood to charge for all, input and working and output tape; hence sequences of reals and real vectors computable in polynomial space also satisfy exponential bounds of growth of value and dimension.

In this way our real counterparts of P and PSPACE above agree with both the conception of real numbers as ‘streams’ of approximations [31,32] as well as with the oracle-based approach [15,9].

Recall that the discrete complexity class #P consists of all total functions $\psi : \{0, 1\}^* \rightarrow \mathbb{N}$ such that some non-deterministic polynomial-time Turing machine on input $\vec{x} \in \{0, 1\}^*$ has precisely $\psi(\vec{x})$ accepting computations.

Definition 6. Fix $\psi : \{0, 1\}^* \rightarrow \mathbb{N}$.

- a) We say that ψ counts the real number $r \in \mathbb{R}$ if it holds $a_n = \psi(1^n) - \psi(0^n)$ according to Definition 4a. If $\psi \in \#P$, call r computable in #P.
b) Say that ψ counts the real sequence $\vec{r} = r_k$ if $a_{n,k} = \psi(1^n 0^k) - \psi(0^n 1^k)$ according to Definition 4c. If $\psi \in \#P$, call $\vec{r}$ computable in #P.

As in Definitions 4, the case of vectors of exponential dimension is more subtle:

Definition 7. a) Suppose $\log(D_k)$ grows at most polynomially. We say that ψ counts the sequence $\vec{r}_k$ of D_k -dimensional real vectors if

$$a_{n,k,J} = \psi(1^n 0^k 1 \text{ bin}(J)) - \psi(0^n 1^k 0 \text{ bin}(J))$$

according to Definition 5a. If $\psi \in \#P$, call $\vec{r}_k$ computable in #P.

- b) Say that ψ counts the partial function $f : \subseteq \mathbb{R} \rightarrow \mathbb{R}$ if

$$b_n(a) = \psi(1^n 0 \text{ bin}(a)) - \psi(0^n 1 \text{ bin}(a))$$

according to Definition 4d. If $\psi \in \#P$, call f computable in #P.

- c) Suppose $\log(D_k)$ grows at most polynomially. We say that ψ counts the sequence of partial vector functions $\vec{f}_k : \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^{D_k}$ if

$$b_{n,k,J}(a) = \psi(1^n 0^k \text{ bin}(J), \text{bin}(a)) - \psi(0^n 1^k \text{ bin}(J), \text{bin}(a))$$

according to Definition 5b. If $\psi \in \#P$, call $\vec{f}_k$ computable in #P.

It is unknown under which arithmetic operations $\#P$ is closed; for instance GapP has been introduced as the closure of $\#P$ under subtraction [6]. The above real counterparts could thus perhaps more accurately be called “ GapP -computability”, rather than $\#P$ -computability. Since it holds $P^{\#P} = P^{\text{GapP}}$, the difference between $\#P$ and GapP seems minor, if any, from the perspective of computational cost. For notational convenience, we define *real counting complexity* neglecting such subtleties. Also Definition 4a) actually refers to P_1 rather than P . Similarly, Definition 6a) could perhaps better (but more awkwardly) refer to “ $\#P_1$ -computability”, since it employs restrictions $\psi|_{1^*}$ of $\psi \in \#P$ to *unary* arguments. This might cause our notions to slightly deviate from [15, p.184].

Polynomial-time computability implies computability in $\#P$, which in turn implies computability in PSPACE : Note that, since $\psi \in \#P$ grows in value at most exponentially in the input length, any real sequence r_k computable in $\#P$ according to c) also has $|r_k|$ growing at most exponentially in k .

2 Techniques and Proof Ideas

For the difference scheme approach [6], taking into account Hypotheses 1 (ii), (iii), the computational problem is equivalent to: first raise a matrix $\mathbf{A}$ (or, rather, any desired one from a sequence of matrices $\mathbf{A}_{h(n)}$) of exponential dimension $K \sim 2^{O(n)}$ to an exponential power $M \sim 2^{O(n)}$; then multiply the intermediate result to a K -dimensional sample vector $\varphi^{(h)}$ of the initial condition φ ; and finally return an approximation to any desired entry with given index $\#J$ of the result vector $(\mathbf{A}_{h(n)}^{M(n)} \cdot \varphi^{(h(n))})_J$ up to error $1/2^n$, $0 \leq J < K$. Naïvely the intermediate result matrices and vectors have exponential dimension, hence leading to complexity class EXP .

On the other hand, according to Question 1, only one entry $\#J$ of the result is required; and both the difference-scheme matrix $\mathbf{A}_{h(n)}$ and the sampled initial vector $\varphi^{(h(n))}$ do not need to be stored, but by hypothesis any desired of its entries can be (re-)computed on-the-fly in P , whenever and however often required: only $J \in \{0, 1, \dots, K-1\}$ and M are part of the input, given in binary with a linear number $\mathcal{O}(n)$ of bits.

The discrete counterpart to our real problem would ask for any desired entry of a P -computable Boolean matrix of exponential dimension to exponential power; and this can be solved in PSPACE by Savitch’s Algorithm. When applying the same approach to the integer or to the present real case, the hypothesis of $\mathbf{A}$ having bounded powers (Definition 2b) becomes crucial: to guarantee that the resulting entries do not blow up, nor do they require excessive initial precision in order to keep the rounding error propagation in check (details omitted). We thus have an improved, namely PSPACE algorithm, and establish Theorem 2a).

Regarding Theorem 2b), Subsection 2.2 reduces the problem of (recovering any desired entry of a) circulant matrix raised to an exponential power to that of (recovering any desired coefficient of a) polynomials raised to such power; in a

way that relates bandwidth to degrees. Subsection 2.1 solves the latter problem in $\#P$ by reduction to Riemann integration via Cauchy's Differentiation Theorem. Linear polynomials can even be raised to exponential powers in P according to Proposition 1.

Subsection 2.3 concludes the proof of Theorem 2 a), b) by treating both grid and nongrid points to compute the solution of (1); this proof makes use of Hypotheses 1 (i). Subsections 2.4, 2.5 establish Theorem 2 c), d), respectively.

2.1 Raising Polynomials to Exponential Powers

Consider the problem of raising a fixed P -computable univariate polynomial P of 'small' degree to an exponential M -th power, and read off the J -th coefficient $P[X^J]$ with $J, M \in \mathbb{N}$ in binary as only actual input. Of course having bounded powers (Definition 3c) is again a crucial hypothesis to prevent coefficient explosion and numerical instability. Note that already for the quadratic polynomial $P(X) = (1 + X + X^2)/3$, naïve evaluation of the 'explicit' formula

$$\left(\frac{1}{3} + \frac{1}{3}X + \frac{1}{3}X^2\right)^K [X^J] = 3^{-K} \cdot \sum_{\substack{0 \leq \mu, \nu \leq K \\ \mu + 2\nu = J}} \frac{K!}{\mu! \cdot \nu! \cdot (K - \mu - \nu)!} \quad (8)$$

involves terms like $K!$ of value, and the sum with a number of terms, doubly exponential in k : not at all obvious to compute in $\#P$.

However, thanks to Cauchy's Integral Theorem, we can express any single desired coefficient of P^M as loop integral over $P^M(z)/z^{M+1}$ for $|z| = 1$ running over the complex unit circle. And due to P having bounded powers, the values of $P^M(z)/z^{M+1}$ are bounded, can be computed with repeated squaring on real numbers, and their dependence on z is sufficiently well-behaved: Such Riemann integrals are known computable in $\#P$ [15, §5.4]. This generalizes to the multivariate case:

Theorem 3. *Fix a sequence $P_k(\vec{X})$ of L -variate polynomials of componentwise degree $< \vec{D}$ and bounded powers. Let K_k denote a sequence of natural numbers with binary representation computable in time polynomial (and thus of value exponentially bounded) in k .*

Then each coefficient $P_k^{K_k}[\vec{X}^{\vec{j}}]$ of $P_k^{K_k}$, $\vec{0} \leq \vec{j} < \vec{D}$, formalized as mapping $\text{bin}(1^k, \text{bin}(\vec{j})) \mapsto P_k^{K_k}[\vec{X}^{\vec{j}}]$, is computable in $\#P$.

Note that $P_k^{K_k}$ has componentwise degree $< \vec{D} \cdot K^k$, increased at most exponentially in k ; hence the binary length of $\vec{j}$ is polynomial in k , the binary length of 1^k . Recall Definition 6 for the formal notion of a (sequence/family of) real vectors/matrices/polynomials to be computable in $\#P$. Theorem 3 extends immediately to Laurent polynomials.

Raising linear polynomials is feasible in P : Note that $\binom{N}{K} p^K q^{N-K}$ is the coefficient of X^K of the polynomial $(pX + q)^N$.

Proposition 1. *Fix P -computable $q \in (0; 1)$ and $k \in \mathbb{N}$. Abbreviate $p := 1 - q$. Given $K \leq N \in \mathbb{N}$, one can approximate $\binom{N}{K} p^K q^{N-K}$ to absolute error N^{-k} in time polynomial in $\log N$.*

2.2 Circulant Matrices as Polynomials

Consider an $N \times N$ circulant matrix $\mathcal{A}$ with parameters $a_0, \dots, a_{N-1}$,

$$\mathcal{A} = (a_{I-J \bmod N})_{0 \leq I, J < N} = \sum_{J=0}^{N-1} a_J \cdot \mathcal{C}_N^J \quad (9)$$

for the ‘generator’ circulant matrix $\mathcal{C}_N$ with parameters $(0, 1, 0, 0, \dots, 0)$. Let $P_{\mathcal{A}}(X) = \sum_{j=0}^{N-1} a_j X^j$ denote the *associated polynomial* to circulant matrix $\mathcal{A}$: so $\mathcal{A}^M = P_{\mathcal{A}}^M(\mathcal{C}_N)$, and the entry with indices (I, J) of $\mathcal{A}^M$ is

$$(\mathcal{A}^M)_{I,J} = \sum_{K \equiv (I-J) \bmod N} P_{\mathcal{A}}^M[X^K]$$

in the terminology of Definition 3. The associated polynomial is a well-known concept [18]. We will primarily consider circulant matrices A with small bandwidth D . These correspond to the images of *Laurent* polynomials with small degrees D : linear combinations of monomials X^j of possibly negative exponents j ranging from $-D$ to $+D$. More generally, we treat matrices that can be decomposed into components combined by Kronecker product as in Equation (11).

Lemma 1. *Fix a not necessarily commutative ring $\mathcal{R}$ of characteristic zero. Let $\text{CIRC}_D^N(\mathcal{R}) \subseteq \mathcal{R}_D^{N \times N}$ denote the subspace of $N \times N$ circulant matrices of bandwidth $< D$. Recall that $\mathcal{C}_N \in \text{CIRC}_2^N(\mathcal{R})$ denotes the $N \times N$ cyclic permutation matrix from (9) and (Definition 2) we write $P[\vec{X}^{\vec{j}}] \in \mathcal{R}$ for the coefficient to $\vec{X}^{\vec{j}} = X_1^{j_1} \cdots X_L^{j_L}$ in an L -variate Laurent polynomial $P \in \mathcal{R}[\vec{X}, \vec{X}^{-1}]$, $\vec{j} \in \mathbb{Z}^L$.*

a) *For any $P \in \mathcal{R}_D[X, X^{-1}]$ it holds $P(\mathcal{C}_N) \in \text{CIRC}_D^N(\mathcal{R})$. More precisely*

$$P(\mathcal{C}_N)_{I,J} = \sum_{n \in \mathbb{Z}} P[X^{J-I+nN}] \in \mathcal{R} ,$$

and the sum is finite. The mapping $\mathcal{R}[X, X^{-1}] \ni P \mapsto P(\mathcal{C}_N) \in \text{CIRC}^N(\mathcal{R})$ is a homomorphism of non-commutative algebras. For $D \leq N/2$ and normed $\mathcal{R}$, the restriction $\mathcal{R}_D[X, X^{-1}] \ni P \mapsto P(\mathcal{C}_N) \in \text{CIRC}_D^N(\mathcal{R})$ is an isometry of vector spaces with respect to the induced norms from Definitions 3 and 2.

b) *Generalizing a), fix $m, L, N_1, \dots, N_L \in \mathbb{N}$ and consider ring homomorphism*

$$\begin{aligned} \Phi_{m, \vec{N}} : \mathbb{R}^{m \times m}[X_1, X_1^{-1}, \dots, X_L, X_L^{-1}] &\ni P \mapsto \mathbb{R}^{m \times m} \otimes \mathbb{R}^{N_1 \times N_1} \otimes \dots \otimes \mathbb{R}^{N_L \times N_L} \\ \mathbb{R}^{m \times m} \ni \mathbf{B} &\mapsto \mathbf{B} \otimes \mathbb{D}_{N_1} \otimes \dots \otimes \mathbb{D}_{N_L}, \quad X_\ell \mapsto \mathbb{D}_m \otimes \mathbb{D}_{N_1} \otimes \dots \otimes \mathcal{C}_{N_\ell} \otimes \dots \otimes \mathbb{D}_{N_L} \end{aligned}$$

Then it holds $(\Phi_{m, \vec{N}}(P))_{I_1, J_1, \dots, I_L, J_L} =$

$$\sum_{n \in \mathbb{Z}} P[X_1^{J_1 - I_1 + n_1 N_1}, \dots, X_L^{J_L - I_L + n_L N_L}] \in \mathbb{R}^{m \times m}. \quad (10)$$

For $D_\ell \leq N_\ell/2$, $\Phi_{m, \vec{N}}$ restricted to $\mathbb{R}_D^{m \times m}[\vec{X}, \vec{X}^{-1}]$ maps isometrically and surjectively onto the vector space of matrices of the form (11). $\square$

The following Theorem [4] now is a corollary to Theorem [3]: Translate the matrix $\mathbf{A} = \mathbf{A}_k$ into normal form according to Lemma [1b]; then further on into a multivariate Laurent polynomial P according to the inverse isometry from Lemma [1c]; raise P_k^M to the desired power following Theorem [3] and finally recover the desired coefficient of $\mathbf{A}^M$ according to Equation [10].

Theorem 4. Fix $J, L \in \mathbb{N}$ and, for $k = 1, 2, \dots$, let

- $K = K(k, j, \ell) \in \mathbb{N}$ with binary representation computable in time polynomial (and thus of value exponentially bounded) in k , $1 \leq j \leq J$, $1 \leq \ell \leq L$
- $\mathcal{C}_{k,j,\ell}$ polynomial-time computable circulant matrices of bandwidth $< D$ and dimensions $K = K(k, \ell)$, $\ell \leq L$
- Q_j polynomial time computable real matrices of constant dimension d , $1 \leq j \leq J$
- $M = M(k)$ natural numbers with binary representations computable in time polynomial in k .

Consider the following (sequence of) Kronecker/tensor products:

$$\mathbf{A}_k := \sum_{j=1}^J Q_j \otimes \mathcal{C}_{k,j,1} \otimes \mathcal{C}_{k,j,2} \otimes \cdots \otimes \mathcal{C}_{k,j,L} \quad (11)$$

If that matrix sequence has bounded powers, then the sequence $\mathbf{A}_k^{M(k)}$ of matrix powers is computable in $\#P$. $\square$

Replacing Theorem [3] with Proposition [1], we obtain the following corollary which may be of independent interest:

Corollary 1. If the circulant matrices $\mathcal{C}_{k,j,\ell}$ in Theorem [4] have bandwidth two, then the sequence $\mathbf{A}_k^{M(k)}$ of matrix powers is computable in polynomial time.

Difference schemes of bandwidth two correspond to simple transport equations, which admit explicit solutions in Subsection [2.4]. Also, fast exponentiation A^M of a difference scheme A by itself is insufficient: It remains to apply the inner product with the initial condition, which still incurs cost $\#P$ [17, Proposition 6e].

2.3 Complexity of Solutions of Evolutionary PDEs

Proof (Theorem [2a,b]). To compute the solution $\mathbf{u}$ at a fixed point (t, x) with the prescribed precision 2^{-n} and estimate the bit-cost of the computation, consider the following computation steps.

1. Choose the space and time grid steps, h and τ , in the following way:
 - h is any binary-rational number of the form $h = 2^{-N}$, where $N = \mathcal{O}(n)$, satisfying the Inequality [13] below;
 - τ is any binary-rational number meeting the Courant inequality $\tau \leq \nu h$, where ν is the Courant number guaranteeing convergence property of the considered difference scheme which can be computed from the coefficients of the system [1].

2. For a grid point $(t, \vec{x})$ put $l = \frac{t}{\tau}$ (note that $l \leq M = \lceil \frac{1}{\tau} \rceil = \mathcal{O}(2^n)$) and calculate the matrix powers and vector products $\mathbf{A}_{h(n)}^{M(n)} \cdot \varphi^{(h(n))}$. Note that we use matrix powering instead of step-by-step iterations initially suggested by the difference scheme (6).

3. For non-grid points take (e.g.) a multilinear interpolation $\widetilde{u^{(h)}}$ of $u^{(h)}$, which can be computed in polynomial time from the (constant number of) “neighbor” grid points. Due to well-known properties of multilinear interpolations,

$$\sup_{t, \vec{x}} |\widetilde{u^{(h)}}(t, \vec{x})| \leq \tilde{C} \sup_{G_N^\tau} |u^{(h)}|; \quad \sup_{t, \vec{x}} |\mathbf{u}(t, \vec{x}) - \widetilde{\mathbf{u}}|_{G_N^\tau}(t, \vec{x})| \leq \tilde{C} \sup_{t, \vec{x}} |\partial^{\vec{2}} \mathbf{u}(t, \vec{x})| \cdot h^2, \quad (12)$$

where $\tilde{C}$ and $\bar{C}$ are absolute constants.

Based on (12) and on the continuous dependence property, as well as on linearity of the interpolation operator, infer

$$\begin{aligned} \sup_{t, \vec{x}} |\mathbf{u}(t, \vec{x}) - \widetilde{u^{(h)}}(t, \vec{x})| &\leq \sup_{t, \vec{x}} \left(|\mathbf{u}(t, \vec{x}) - \widetilde{\mathbf{u}}|_{G_N^\tau}(t, \vec{x})| + |\widetilde{\mathbf{u}}|_{G_N^\tau}(t, \vec{x}) - \widetilde{u^{(h)}}(t, \vec{x})| \right) \\ &\leq \bar{C} C_0 \|\varphi\|_{C^1(\bar{\Omega})} h^2 + \tilde{C} C \cdot h \leq 2^{-n}. \end{aligned}$$

Thus choosing a grid step $h = 2^{-N}$ such that

$$h \leq C_h \cdot 2^{-n}, \quad C_h = \bar{C} C_0 \|\varphi\|_{C^1(\bar{\Omega})} + \tilde{C} C, \quad (13)$$

will guarantee the computed function $\widetilde{u^{(h)}}$ approximate the solution u with the prescribed precision 2^{-n} (here C_h depends only on the fixed P-computable functions φ , B_i and therefore is a fixed constant. Hence it remains to estimate complexity of matrix powering (which turns out to be in PSPACE or #P in items a) or b) of Theorem 2, respectively) and inner product (which turns out to be in #P), see comments in the beginning of this Section and Theorem 3.

2.4 Symmetric Hyperbolic PDEs with Commuting Coefficients

Towards proving Theorem 2c), first consider the following *scalar* linear PDE with constant coefficients and its explicit solution

$$\lambda_0 u_t = \sum_j \lambda_j \partial_j u, \quad u(\vec{x}, t) = u_0(t \sum_j \lambda_j + \lambda_0 \sum_j x_j) \quad (14)$$

which is obviously computable in P, provided that $\lambda_0, \lambda_1, \dots, \lambda_d \in \mathbb{R}$ are.

Theorem 2c) considers *vector systems* of such PDEs, with constant symmetric mutually commuting matrix coefficients $\mathbf{B}_{\vec{j}}$. These hypotheses assert a simultaneous diagonalization, that is, a basis of joint eigenvectors: which ‘decouples’ the system into e independent scalar equations (14). It remains to prove that such a joint spectral decomposition can be computed in P: which is wrong in case (even a single symmetric) matrix $\mathbf{B}_{\vec{j}}$ is given as input [34]. Fortunately, in agreement with Question 1, these matrices are not part of the input but fixed computable in P. And for this case we have and apply the following

Theorem 5. *Fix symmetric and mutually commuting matrices $\mathbf{B}_{\vec{j}} = \mathbf{B}_{\vec{j}}^*$ of fixed dimension $e \in \mathbb{N}$ with fixed real polynomial-time computable entries. Then a joint spectral decomposition $\mathbf{B}_{\vec{j}} = \mathbf{T}^\top \cdot \mathbf{D}_{\vec{j}} \cdot \mathbf{T}$ can be computed in time polynomial in the output precision parameter n (and unspecified dependence on the dimension e as well as on the ‘input’ matrix entries). Here $\mathbf{D}_{\vec{j}}$ is a diagonal matrix with the eigenvalues of $\mathbf{B}_{\vec{j}}$; and $\mathbf{T}$ is an orthonormal matrix consisting of a joint basis of eigenvectors of all $\mathbf{B}_{\vec{j}}$.*

Proof. Let k denote the dimension of B . By [21], the d -tuple of eigenvalues with multiplicity (i.e. the sought diagonal matrix D) can be approximated up to absolute error $1/2^n$ in time polynomial in n . Note that the (integer) multiplicities themselves depend discontinuously on the entries of B and therefore cannot be computed; but, since the matrix is fixed, they can be hardcoded into the thus non-uniform computation. It then remains to compute, for each eigenvalue λ with correct multiplicity $k = k(\lambda)$, an orthonormal basis of the k -dimensional kernel of $B - \lambda \cdot \text{id}$ [23]. Naive Gaussian Elimination does not suffice for this purpose since the tests for in/equality during pivot search are undecidable. Bareiss Algorithm avoids the divisions; but cannot avoid tests either. On the other hand Bareiss performs (more than Gauss, but still) only a number of test polynomially bounded in the dimension d , which is constant in our setting, namely independent of the output precision parameter n . Hence, again, the outcomes of these tests can be hardcoded in this non-uniform computation. The remaining operations of Bareiss are arithmetic, and can be performed on approximations. More precisely consider the $d \times d$ matrix $\tilde{B}_{m,\lambda}$ of integer numerators such that dyadic $\tilde{B}_{m,\lambda}/2^m$ approximates $B - \lambda \cdot \text{id}$ ‘sufficiently’ well: Since the number of arithmetic operations is polynomial in d , it suffices to choose m polynomially larger than n in order to guarantee that the output of Bareiss is still within absolute error $1/2^n$ from the hypothetical exact result. The bit-cost of Bareiss is well-known polynomial in $(d \text{ and}) m$.

2.5 #P₁ is Optimal for the Heat Equation

#P is known optimal for *indefinite* Riemann Integration [15, Theorem 5.33], while #P₁ is optimal for *definite* Riemann Integration [15, Theorem 5.32] in the following sense:

- Fact 6** a) *There is a P-computable (hence continuous) $h : [0; 1] \rightarrow [-1; 1]$ such that $\int_0^1 h(x) dx$ is not computable in P₁ unless $\text{FP}_1 = \text{\#P}_1$.*
b) *For every P-computable analytic function $g : [0; 1] \rightarrow \mathbb{R}$, $\int_0^1 g(x) dx$ is computable in P₁ [15, bottom of page 208].*
c) *Let $u = u(x, t)$ solve the 1D Heat Equation $u_t = u_{xx}$ on $[0; 1]$ with periodic boundary conditions $u(0, t) = u(1, t)$ and $u_x(0, t) = u_x(1, t)$ such that $u_0 = u(x, 0)$ is in $\mathcal{C}^2[0; 1]$. Then $u(x, t)$ is analytic in x for each fixed $t > 0$, and the ‘overall heat’ $\int_0^1 u(x, t) dx$ does not depend on t ; see e.g. [7].* $\square$

In order to establish Theorem [21], consider $u_0 := h$ from Fact [6a). W.l.o.g. $h(0) = h(1)$: otherwise consider $[0; 1/2] \ni x \mapsto h(2x)/2$ and $[1/2; 1] \ni x \mapsto h(2 - 2x)/2$. Then apply Fact [6b) to $g(x) := u(x, t)$ according to Fact [6c).

3 Conclusion and Perspective

We have extended previous rigorous bit-cost investigations of PDEs [13] from elliptic to hyperbolic and parabolic linear PDEs, improving upper complexity bounds PSPACE and $\#P^{\#P}$ [17] to $\#P$. $\#P_1$ (not $\#P$, as claimed in [13, §6]) turned out as necessary for both Poisson and Heat equation. Current (limited) evidence suggests that differential equations might exhibit a dichotomy [26]: either P (Subsection 2.4 of the present paper, Theorem 3 of [17] for the case of linear evolutionary systems of PDEs with *analytic* initial functions and matrix coefficients, and [1] for analytic ODEs) or $\#P/\#P_1$ -hard.

References

1. Bournez, O., Graça, D.S., Pouly, A.: Solving analytic differential equations in polynomial time over unbounded domains. In: Mathematical foundations of computer science 2011. Lecture Notes in Comput. Sci., vol. 6907, pp. 170–181. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22993-0_18
2. Braverman, M., Cook, S.: Computing over the reals: Foundations for scientific computing. Notices of the AMS **53**(3), 318–329 (2006)
3. Evans, L.: Partial Differential Equations, Graduate Studies in Mathematics, vol. 19. American Mathematical Society (1998)
4. Godunov, S., Ryaben’kii, V.: Difference Schemes: An Introduction to the Underlying Theory, Studies in Mathematics and Its Applications, vol. 19. North-Holland (1987)
5. Gordienko, V.: Hyperbolic systems equivalent to the wave equation. Siberian Mathematical Journal **50**, 14–21 (2009)
6. Hemachandra, L.A., Ogiwara, M.: Is $\#P$ closed under subtraction? Current Trends in Theoretical Computer Science pp. 523–536 (1993). https://doi.org/10.1142/9789812794499_0039
7. John, F.: Partial Differential Equations. Springer-Verlag New York, (1982)
8. Kawamura, A.: Lipschitz continuous ordinary differential equations are polynomial-space complete. Computational Complexity **19**(2), 305–332 (2010)
9. Kawamura, A., Cook, S.: Complexity theory for operators in analysis. ACM Trans. Comput. Theory **4**(2), 5:1–5:24 (May 2012)
10. Kawamura, A., Ota, H., Rösnick, C., Ziegler, M.: Computational complexity of smooth differential equations. Logical Methods in Computer Science **10**, 1:6,15 (2014). [https://doi.org/10.2168/LMCS-10\(1:6\)2014](https://doi.org/10.2168/LMCS-10(1:6)2014)
11. Kawamura, A., Steinberg, F., Thies, H.: Parameterized complexity for uniform operators on multidimensional analytic functions and ODE solving. In: Proc 25th Int. Workshop on Logic, Language, Information, and Computation (WOLLIC). pp. 223–236 (2018). https://doi.org/10.1007/978-3-662-57669-4_13
12. Kawamura, A., Steinberg, F., Ziegler, M.: Complexity theory of (functions on) compact metric spaces. In: Proc. 31st Ann. Symposium on Logic in Computer Science, LICS. pp. 837–846. ACM (2016). <https://doi.org/10.1145/2933575.2935311>
13. Kawamura, A., Steinberg, F., Ziegler, M.: On the computational complexity of the Dirichlet problem for Poisson’s equation. Mathematical Structures in Computer Science **27**:8, 1437–1465 (2017). <https://doi.org/10.1017/S096012951600013X>

14. Kawamura, A., Thies, H., Ziegler, M.: Average-case polynomial-time computability of hamiltonian dynamics. In: Potapov, I., Spirakis, P.G., Worrell, J. (eds.) 43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK. LIPIcs, vol. 117, pp. 30:1–30:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018)
15. Ko, K.I.: Complexity Theory of Real Functions. Progress in Theoretical Computer Science, Birkhäuser, Boston (1991)
16. Kolmogorov, A.N., Tikhomirov, V.M.: E-entropy and E-capacity of sets in functional spaces. In: Shiryayev, A. (ed.) Selected Works of A.N. Kolmogorov, vol. III, pp. 86–170. Springer (1993), originally pp.3–86 in *Uspekhi Mat. Nauk* vol.**14**:2 (1959)
17. Koswara, I., Selivanova, S., Ziegler, M.: Computational complexity of real powering and improved solving linear differential equations. In: Proc. 14th International Computer Science Symposium in Russia. LNCS, vol. 11532 (2019)
18. Kra, I., Simanca, S.R.: On circulant matrices. Notices of the AMS **59**(3), 368–377 (2012). <https://doi.org/10.1090/noti804>
19. LeVeque, R.J. (ed.): Finite Difference Methods for Differential Equations. SIAM, Philadelphia (2007)
20. Liao, H.L., Sun, Z.z.: Maximum norm error estimates of efficient difference schemes for second-order wave equations. Journal of Computational and Applied Mathematics **235** (8), 2217–2233 (2011). <https://doi.org/10.1016/j.cam.2010.10.019>
21. Neff, C.A.: Specified precision polynomial root isolation is in NC. J. Comput. Syst. Sci. **48**(3), 429–463 (1994). [https://doi.org/10.1016/S0022-0000\(05\)80061-3](https://doi.org/10.1016/S0022-0000(05)80061-3)
22. Pan, V., Reif, J.: The bit-complexity of discrete solutions of partial differential equations: Compact multigrid. Computers and Mathematics with Applications **20**, 9–16 (1990)
23. Park, S.: Reliable degenerate matrix diagonalization. Tech. Rep. CS-TR-2018-415, KAIST (2018)
24. Park, S., Brauße, F., Collins, P., Kim, S., Konečný, M., Lee, G., Müller, N., Neumann, E., Preining, N., Ziegler, M.: Foundation of computer (algebra) analysis systems: Semantics, logic, programming, verification. <https://arxiv.org/abs/1608.05787> (2020)
25. Pour-El, M.B., Richards, J.I.: The wave equation with computable initial data such that its unique solution is not computable. Advances in Math. **39**, 215–239 (1981)
26. Schaefer, T.J.: The complexity of satisfiability problems. In: Lipton, R.J., Burkhard, W.A., Savitch, W.J., Friedman, E.P., Aho, A.V. (eds.) Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA. pp. 216–226. ACM (1978)
27. Selivanova, S., Selivanov, V.L.: Computing solution operators of boundary-value problems for some linear hyperbolic systems of PDEs. Logical Methods in Computer Science **13**(4) (2017). [https://doi.org/10.23638/LMCS-13\(4:13\)2017](https://doi.org/10.23638/LMCS-13(4:13)2017)
28. Selivanova, S.V., Selivanov, V.L.: Bit complexity of computing solutions for symmetric hyperbolic systems of PDEs (extended abstract). In: Proceedings of CiE 2018. pp. 376–385 (2018). https://doi.org/10.1007/978-3-319-94418-0_38
29. Strikwerda, J.C.: Finite Difference Schemes and Partial Differential Equations. SIAM (2004)
30. Sun, S., Zhong, N., Ziegler, M.: Computability of the solutions to Navier-Stokes equations via effective approximation. In: Du, D., Wang, J. (eds.) Complexity and Approximation - In Memory of Ker-I Ko. Lecture Notes in Computer Science, vol. 12000, pp. 80–112. Springer (2020). https://doi.org/10.1007/978-3-030-41672-0_7

31. Weihrauch, K.: *Computable Analysis*. Springer, Berlin (2000)
32. Weihrauch, K.: Computational complexity on computable metric spaces. *Mathematical Logic Quarterly* **49:1**, 3–21 (2003)
33. Weihrauch, K., Zhong, N.: Is wave propagation computable or can wave computers beat the Turing machine? *Proceedings of the London Mathematical Society* **85**(2), 312–332 (2002)
34. Ziegler, M., Brattka, V.: A computable spectral theorem. In: Blanck, J., Brattka, V., Hertling, P. (eds.) *Computability and Complexity in Analysis*. *Lecture Notes in Computer Science*, vol. 2064, pp. 378–388. Springer, Berlin (2001), 4th International Workshop, CCA 2000, Swansea, UK, September 2000