REVISTA MATEMÁTICA de la
Universidad Complutense de Madrid
Volumen 10, número Suplementario: 1997

Semi-algebraic complexity-additive complexity of diagonalization of quadratic forms.

Thomas LICKTEIG[†] and Klaus MEER[‡]

Abstract

We study matrix calculations such as diagonalization of quadratic forms under the aspect of additive complexity and relate these complexities to the complexity of matrix multiplication. While in [BKL] for multiplicative complexity the customary "thick path existence" argument was sufficient, here for additive complexity we need the more delicate finess of the real spectrum (cf. [BCR], [Be], [KS]) to obtain a complexity relativization. After its outstanding success in semi-algebraic geometry the power of the real spectrum method in complexity theory becomes more and more apparent. Our discussions substantiate once more the signification and future rôle of this concept in the mathematical evolution of the field of real algebraic algorithmic complexity.

A further technical tool concerning additive complexity is the structural transport metamorphosis from [Li1] which constitutes another use of the exponential and the logarithm as it appears in the work on additive complexity by [Gr] and [Ri] through the use of [Kh].

We confine ourselves here to diagonalization of quadratic forms. In the forthcoming paper [LM] further such relativizations of additive complexity will be given for a series of matrix computational tasks.

[†]Supported by DFG Heisenberg Grant Li-405/2-1
[‡]Partially supported by EC working group NeuroCOLT 8556
AMS(MOS) subject classification: 68Q05, 68Q25, 68Q40; 14P10, 14P20

1 Introduction

We start our discussion with some general remarks of thematizing character. The question for the complexity of computing the value $y = f(x)$ of a polynomial (say) function $f : \mathbb{R}^n \to \mathbb{R}^m$ is always accompanied by the question for the complexity of checking correctness of $y = f(x)$ given $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$. It can be reasonably conjectured that in many cases no big complexity differences appear, and likewise as well, that in several cases the latter decision task may be of considerably lower complexity than the computation task; however no systematic studies do exist so far. We refer to [Schö3] for a discussion of this general *computation versus verification* dualism.

For instance, computing the solution $y = X^{-1}x = f(X, x)$ of a linear system, X a regular square matrix, seems to be more difficult than the obvious matrix times vector "control calculation" for $Xy \stackrel{?}{=} x$. This phenomenon has been transfered into practical utilization since a long time in the relaxed form of the numerical *a posteriori* check $||Xy - x|| \stackrel{?}{\leq} \epsilon$. If additional information about the condition number of the matrix is available, testing membership in this tube is useful for repairing an *ignorabimus*, that is, making numerical solution algorithms "waterproof" whenever a priori bounds on the error propagation are – or are expected to be – too pessimistic. On the other hand side, original numerical computation requirements in the sense of a backward analysis elucidate the *a priori* task of computing some output such that the input-output pair lies in the tube. Similarly – depending on the perspective – also in other cases, producing *some* output plus a definite information rather than a specific function value $y = f(x)$ constitutes the computational task. We mention furthermore that also the fundamental and important question for the possibilities of algorithmic savings by relaxations through the introduction of *redundant representations* of (intermediate) results belongs within the scope of this theme just outlined. The most simple example is the interpretation of the *carry save adders* in Boolean parallel complexity as the relaxation of integer addition $y = x + x'$ to the task of computing $(y, y') \in \mathbb{Z}^2$ given $(x, x', x'') \in \mathbb{Z}^3$ with $y + y' = x + x' + x''$; the question also plays a very substantial rôle in elimination theory (see [HM], [GHMP] and the references given there).

Let R be a real closed field. A *semi-algebraic computational task of*

format $(n;m)$ is given by a semi-algebraic *input-output specification* $\mathcal{S}$, that is a semi-algebraic subset

$$\mathcal{S} \subseteq R^n \times R^m. \tag{1}$$

(The idea of such a definition of a task specification appears already in [vNG], [Tu], and recently in the papers [Schö3], [BSS], [Li1]). $\mathcal{S}$ – i.e., the underlying semi-algebraic subset plus the given split – can be considered as a semi-algebraic family of subsets of R^m indexed by R^n (cf. [BCR], Chapitre 7.4); its *domain* $\mathcal{D}(\mathcal{S})$ is the subset of all $x \in R^n$ such that the fiber

$$\mathcal{S}_x = \{y \in R^m : (x, y) \in \mathcal{S}\}$$

is not empty.

Let $\Omega^R = R \sqcup \{0, 1, +, -, *, /\}$ and $P = \{=, \leq\}$. Assume $\mathcal{T}$ to be a semi-algebraic computation tree (that is, with output instruction) of operational-relational signature (Ω^R, P) and of input-output format $(n;m)$. $\mathcal{T}$ is said to *solve the task* $\mathcal{S}$ if $\mathcal{T}$ is executable on all $x \in \mathcal{D}(\mathcal{S})$ and its output $\mathcal{T}(x) \in R^m$ on input $x \in R^n$ lies in the fiber $\mathcal{S}_x$ for all $x \in \mathcal{D}(\mathcal{S})$. If this is the case, one can also say that $\mathcal{T}$ computes the *semi-algebraic knowledge* $\mathcal{S}$ *(over* $\mathcal{D}(\mathcal{S})$*)*. Throughout this paper we deal with this *a priori* task.

Such tasks (1) with constant output length m are called *monochrome* in [LM], and only monochrome tasks will appear in the sequel. In general the output length will also vary; then the task is called *polychrome*. See [LM] for a general discussion.

What does "diagonalization of quadratic forms," given as symmetric $m \times m$ matrices S,

$$^tMSM = D = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_m \end{pmatrix}, \tag{2}$$

mean in computational terms? There are three possible interpretations.

- DIAG: Compute some regular $m \times m$ matrix M and diagonal D satisfying (2).
- OGB: Compute some regular M such that there exists a diagonal D with (2).

- Compute some diagonal D such that there exists a regular M with (2).

Irrespective of the cost function considered (additive complexity: $c_+ = 1_{\{+,-\}}$; multiplicative complexity: $c_* = 1_{\{*,/\}}$; total complexity: $c_{\text{tot}} = 1_{\Omega^R \sqcup P}$) the second and the third ones are not harder than the first one, and the third one essentially boils down to the rank-signature decision task. In the sequel we shall give for the second orthogonal basis task and additive complexity a relativization to matrix multiplication. As consequence, the diagonalization task posesses also such a relativization, but a more precise statement on the "difficult input sets" is possible for diagonalization. The fundamental concept of the *real spectrum* ([CR], [Ro], [BCR], [Be], [KS]), introducing great clarity into semi-algebraic geometry and complexity, will again be of exceptional importance in our complexity analysis. (The non-initiate reader may also consult [Li2] for a short summary on this concept).

Section 2 contains complexity notations and tools, section 3 our main lower bound result (Theorem 2) in the semi-algebraic framework which we complement in section 4 with an upper bound discussion in the uniform model of [BSS].

For convenience, $R[X_1, \ldots, X_n]$ and the polynomial functions $\mathcal{P}(R^n)$ on R^n will be identified in several places.

2 Operational and operational-relational complexities, notations and tools

We recall some results and terminology of the complexity modelling from [Li1], [Li2] to be used in later sections. Rings, fields and algebras are always assumed to be commutative.

2.1 Straight line programs – operational complexities

Let R be a ring. $\Omega^R \sqcup \{0, 1, +, -, *, /\}$ is a possible operational signature on the category of R-algebras with the interpretation "division by units" for / and "multiplication with λ" for $\lambda \in R$. Let $R \to A$ be an R-algebra. We consider lists (called points) $x = (x_1, \ldots, x_m) \in \boldsymbol{A}^m_{R \to A}$ with $x_i \in A$. For a cost function $c : \Omega^R \to \mathrm{N}$ and a further $y \in \boldsymbol{A}^n_{R \to A}$, $L(c, x, y)$ denotes the minimum c-length of an Ω^R-SLP Γ over m computing y

from the point x over R. A pair (Γ, x) is called an Ω^R-computation (executable or unexecutable) in the R-algebra $R \to A$; an executable (Γ, x) is said to compute y if all y_i appear in the result sequence $\mathrm{Res}(\Gamma, x)$ of it.

We comment on the notation used. R-algebra morphisms

$$\begin{array}{ccc} & & A \\ & \nearrow & \\ R & & \downarrow \\ & \searrow & \\ & & A' \end{array}$$

transport Ω^R-computations (Γ, x) in $R \to A$ into Ω^R-computations (Γ, x') in the arrival algebra $R \to A'$. (We remark that many lower bound techniques result from an application of a morphism that makes a computational step "unnecessary.") However one needs also the variable point of view with respect to coefficients. So one is lead to the wider category **flac** of *all* commutative algebras [Li2]. Its morphisms are given by commutative squares

$$\begin{array}{ccc} R & \to & A \\ \downarrow & & \downarrow \\ R' & \to & A'. \end{array}$$

They transport points $x \in \boldsymbol{A}^m_{R \to A}$ into points $x' \in \boldsymbol{A}^m_{R' \to A'}$ and Ω^R-computation (Γ, x) in $R \to A$ into $\Omega^{R'}$-computations (Γ', x') in $R' \to A'$ which is the functorial conception of *affine space.*

Remark. As a rule, the morphisms in **flac** that will appear will all be canonical, and the algebra reader will perform the transition of points always "mentally." Nevertheless we shall use the *affine space* notation throughout in order to indicate always in explicit form the algebra in which a list is considered. (The computer science reader will observe that the affine space notation in [Li2] is parallel to constructs in AXIOM.)

Considerations will often start in the R-algebra $R[X]$, and some self-explanatory shorthand notations are useful. For the "vector" of indeterminates in the R-algebra $R[X]$ our *standard notation* is $X = (X_1, \ldots, X_m) \in \boldsymbol{A}^m_{R \to R[X]}$. If this vector is considered in $R(X) = k(\mathfrak{o})$

we write $X(\mathfrak{o}) \in \boldsymbol{A}^m_{R \to R(X)}$ or simply $X(\mathfrak{o})$. Likewise we shall use self-explanatory notations such as $X(\mathfrak{p}) \in \boldsymbol{A}^m_{R \to k(\mathfrak{p})}$ and $X_{\mathfrak{p}} \in \boldsymbol{A}^m_{R \to R[X]_{\mathfrak{p}}}$ for $\mathfrak{p} \in \operatorname{Spec} R[X]$, etc. For $\alpha \in \operatorname{Spec}_r R[X]$, $X(\alpha) \in \boldsymbol{A}^m_{R \to k(\alpha)}$ denotes the image vector of X in the real closure $k(\alpha)$ of the residue field $k(\mathfrak{p})$ with respect to the ordering on $k(\mathfrak{p})$ induced by α; here $\mathfrak{p} \in \operatorname{Spec} R[X]$ denotes the support $\mathfrak{p} = \operatorname{supp} \alpha = \alpha \cap -\alpha$ of the prime cone α.

2.2 Verifying straight line programs – operational-relational complexities

Let $P = \{=, \leq\}$. Adding a number of P-comparison instructions at the beginning and after each computational instruction in an Ω^R-SLP over m we get an (Ω^R, P)-SLP Υ over m; these are called verifiers ([Li2]). Inputs are now ordered field points (over R) $(x, \leq)$ where $x \in \boldsymbol{A}^m_{R \to K}$ and $(K, \leq)$ is an ordered field. Since the real spectrum "encompasses" all evaluations $R[X] \to (K, \leq)$ in ordered fields, consideration of the ordered field points $(X(\alpha), \leq_\alpha)$ for $\alpha \in \operatorname{Spec}_r R[X]$ constitutes no restriction of generality. Notationally, we shall make no distinction between the point $X(\alpha) \in \boldsymbol{A}^m_{R \to k(\alpha)}$ and the ordered field point $(X(\alpha), \leq_\alpha)$. (One can also identify $\alpha \in \operatorname{Spec}_r R[X]$ and the list $X(\alpha)$.)

The complexity $V(c, X, \alpha)$ of verifying $\alpha \in \operatorname{Spec}_r R[X]$ in its halo $\operatorname{hal} \alpha = \{\beta : \beta \subsetneqq \alpha\}$ of generizations is defined as the minimum c-length $L(c, \Upsilon)$ of a verifier Υ over m distinguishing $X(\alpha)$ from every $X(\beta)$, $\beta \subset \alpha$, through the outcomes of the comparison test. We then say that Υ verifies α. A verifier Υ over m being given, $\Gamma(\Upsilon)$ denotes the Ω^R-SLP over m of pure computational steps of Υ.

The operational complexity $I(c, X_\alpha, \alpha)$ of isolating α in its halo of generizations is defined as the minimum c-length of an Ω^R-SLP over m computing for some $n \in \mathrm{N}$ a list $f \in \boldsymbol{A}^n_{R \to R[X]_\alpha}$ from X_α such that

$$\mathcal{Z}(f) = \{\alpha\} \text{ in } \operatorname{Spec}_r R[X]_\alpha.$$

Analogously one defines the isolation complexity $I(c, X_{\mathfrak{p}}, \mathfrak{p})$ of a prime ideal $\mathfrak{p} \in \operatorname{Spec} R[X]$, and one has

$$I(c, X_\alpha, \alpha) \leq I(c, X_{\operatorname{supp} \alpha}, \operatorname{supp} \alpha) \tag{3}$$

for $\alpha \in \operatorname{Spec}_r R[X]$. Isolation complexity provides a lower bound on verification complexity under reasonable assumptions on the counting c under consideration ([Li2]).

2.3 Semi-algebraic computation trees – path selection

We recall that there is a one-to-one correspondence between semi-algebraic subsets $E \subseteq R^m$ and *constructible* sets in $\tilde{E} \subseteq \mathrm{Spec}_r R[X]$; this transition is called *operation tilda* (cf. [BCR]). A prime cone $\alpha \in \tilde{E}$ is called a minimal prime cone of $\tilde{E}$ if it possesses no proper generization within $\tilde{E}$.

Let R be a real closed field and $\mathcal{T}$ be a semi-algebraic computation tree of operational-relational signature (Ω^R, P) over m. We trace the path $\mathcal{T}_\alpha$ in $\mathcal{T}$ followed by the input $X(\alpha)$ for $\alpha \in \mathrm{Spec}_r R[X]$. Assume that $\mathcal{T}$ has input-output format $(m;n)$ and solves the monochrome task $\mathcal{S}$ of the same format. The tilda $\tilde{\mathcal{S}}$ of the specification will provide us with information on the behavior of $\mathcal{T}_\alpha$ on α. $\mathcal{T}_\alpha$ can be considered as a verifier over m, and $\Gamma(\mathcal{T}_\alpha)$ computes the output $\mathcal{T}(\alpha) \in \boldsymbol{A}^n_{R \to k(\alpha)}$ of the tree $\mathcal{T}$ on input $X(\alpha)$. The prime cone $\alpha' \in \mathrm{Spec}_r R[X][Y_1, \ldots, Y_n]$ defined by the input-output list $(X(\alpha), \mathcal{T}(\alpha)) \in \boldsymbol{A}^{m+n}_{R \to k(\alpha)}$ lies then in the tilda $\tilde{\mathcal{S}}$ which we are going to use for the concrete tasks mentioned above.

Secondly, $\mathcal{T}_\alpha$ may verify α in its halo of generizations. This property (for certain α) is a consequence of the specification in case of *decision tasks* or other polychrome tasks ([Li2], [LR]). For monochrome tasks separation properties of prime cones are not a direct consequence of the specification. We shall nevertheless encounter a verification discussion of paths $\mathcal{T}_\alpha$ through *two ways* arguments where one of them will be guaranteed to "to catch the mouse."

For a tree $\mathcal{T}$, $L(c, \mathcal{T}_\alpha)$ denotes the c-length of the path $\mathcal{T}_\alpha$; maximizing over all path lengths we have the c-cost $C(c, \mathcal{T})$ of $\mathcal{T}$. Minimizing $C(c, \mathcal{T})$ over all $\mathcal{T}$ solving the task $\mathcal{S}$, defines the "minimax" complexity

$$C(c, \mathcal{S}) = \min_{\mathcal{T} \text{ for } \mathcal{S}} \max_{\alpha \in \widetilde{\mathcal{D}}(\mathcal{S})} L(c, \mathcal{T}_\alpha)$$

of $\mathcal{S}$.

2.4 Additive complexity and the logarithmic metamorphosis

We first recall the statements of two results from [Li1] concerning additive complexity which will be needed to prove our main lower bound

result (Theorem 2).

For $f \in K = R(X)$ write

$$f(X+Y) = \sum_{I \in \mathbf{N}^n} f_I(X) Y^I \in K[[Y_1, \ldots, Y_n]],$$

and let for $d \in \mathbf{N}$

$$j_d(f) = \sum_{|I| \leq d} f_I(X) Y^I \in K[Y_1, \ldots, Y_n]$$

denote its d-jet. The following Theorem relating additive and multiplicative complexity is based on a *program transformation* (Ω^R-SLPs into Ω^K-SLPs) mainly constructed with the help of an appropriate structural transport ("metamorphosis") via

$$\mathfrak{m} \underset{\log}{\overset{\exp}{\leftarrow\!\!\rightarrow}} 1 + \mathfrak{m},$$

$\mathfrak{m}$ denoting the maximal ideal of $K[Y]/(Y_1, \ldots, Y_n)^{d+1}$, arithmetization by "symbolic jetting," and the classical idea in [St2] (see also [Mo], [Wi]).

Theorem 1. [Li1]. *Let $K = R(X_1, \ldots, X_n)$ and $d \geq 2$. For $f = (f_1, \ldots, f_m) \in \boldsymbol{A}^m_{R \to K}$ additive complexity has the lower bound in terms of multiplicative complexity as*

$$L(c_+, X(0), f) \geq \frac{2}{d(d-1)^2} \cdot L(c_*, Y, j_d(f)) - m - n;$$

here $Y = (Y_1, \ldots, Y_n) \in \boldsymbol{A}^n_{K \to K[Y]}$.

■

Beside this main technical tool our subsequent discussion will require an absolute lower bound on the additive isolation complexity of prime cones of height one. If the support $\mathfrak{p}$ of $\alpha \in \mathrm{Spec}_r R[X]$ has height one then the inequality (3) becomes an equality. Moreover,

$$V(c_+, X, \alpha) \geq I(c_+, X_p, \mathfrak{p}) - 1,$$

and for additive complexity we can use the following Euler derivation bound:

Proposition 1 *[Li1].* Define $\Delta_+(f) = \dim_R(\sum RX_i\partial_i f + Rf)/Rf$ for $f \in R[X]$. Then for every prime principal ideal $\mathfrak{p} = (f) \in \operatorname{Spec} R[X]$,

$$.I(c_+, X_{\mathfrak{p}}, \mathfrak{p}) \geq \Delta_+(f).$$

■

For the discriminant we now bound the value of Δ_+ (the bound is in fact an equality, but we do not need this here).

Lemma 1. *For the discriminant* $\operatorname{dis} \in R[S_{ij} : 1 \leq i \leq j \leq m]$,

$$\Delta_+(\operatorname{dis}) \geq \frac{m(m-1)}{2}.$$

Proof. Let $S \in R[S_{ij} : 1 \leq i \leq j \leq m]^{m \times m}$ be the symmetrix matrix with entry S_{ij} for $1 \leq i \leq j \leq m$, and let its $(m-1) \times (m-1)$ minors be denoted $d_{i;j}$. We show by induction on m that $S_{mm}d_{m;m}$ and the $S_{ij}d_{i;j}$ with $1 \leq i < j \leq m$ are R-linearly independent. This is clear for $m = 1$, so assume $m > 1$. For $1 < i \leq j \leq m$ we have

$$S_{ij}d_{i;j} = S_{11}(S_{ij}d_{1i;1j}) + (\text{an } S_{11}\text{-free term}),$$

and that the $S_{1j}d_{1;j}$ are S_{11}-free for $1 < j \leq m$; hence by induction the asserted linear independence follows.

■

3 Quadratic forms and matrix multiplication

The basic idea for constructing fast (square) matrix multiplication algorithms is to look for "non-commutative algorithms" (computations in non-commutative polynomial algebras) that can be used recursively by block matrix calculation (cf. [St1]). We quickly recall this idea (cf. [Pa], [dG]).

A division free Ω^R-SLP Γ over $r+s$ is said to be "(r,s)-bilinear" if the following holds true: when executed on input $XY = (X_1, \ldots, X_r; Y_1, \ldots, Y_s) \in \boldsymbol{A}^{r+s}_{R \to R[X,Y]}$ then in every multiplication step of Γ the first argument is a linear polynomial in $R[X]$ and the second argument is a linear polynomial in $R[Y]$.

Let Γ be such a bilinear algorithm for $m \times m$ matrix multiplication ($r = s = m^2$) with $n(\Gamma)$ non-linear and $l(\Gamma)$ linear computational steps, assume $m \geq 2$, and let $\tau = \log_m n(\Gamma) > 2$. Then recursively one obtains for $m^k \times m^k$ matrix multiplication a bilinear algorithm, denoted $\Gamma^{(k)}$, satisfying

$$(n+l)(\Gamma^{(k)}) = n(\Gamma) \cdot (n+l)(\Gamma^{(k-1)}) + l(\Gamma) \cdot (m^{k-1})^2$$

for $k > 1$. Dividing both sides by $m^{k\tau}$ shows via a geometric series argument the boundedness of the quotient

$$\begin{aligned} \frac{(n+l)(\Gamma^{(k)})}{m^{k\tau}} &= \frac{(n+l)(\Gamma^{(k-1)})}{m^{(k-1)\tau}} + \frac{l(\Gamma)}{n(\Gamma)} \cdot \frac{m^{(k-1)\cdot 2}}{m^{(k-1)\tau}} \\ &= \frac{(n+l)(\Gamma)}{n(\Gamma)} + \frac{l(\Gamma)}{n(\Gamma)} \cdot \sum_{j=1}^{k-1} m^{(2-\tau)j} \\ &\leq \frac{(n+l)(\Gamma)}{n(\Gamma)} + \frac{l(\Gamma)}{n(\Gamma)} \cdot \frac{m^{2-\tau}}{1-m^{2-\tau}} \quad (\text{by } \tau > 2) \end{aligned}$$

and the irrelevance of the number $l(\Gamma)$ for boundedness itself. Other (square) matrix formats can be augmented with zero blocks to the next $m^k \times m^k$ format.

Let $\mathbf{MAMU} = (\mathrm{MAMU}_m : m \in \mathrm{N}_+)$ denote the sequence of $m \times m$ matrix multiplication computational tasks. The above recursion makes clear the following two things.

1. The *asymptotic exponent* ω of matrix multiplication,

$$\omega = \omega(\mathbf{MAMU}) = \inf\{\tau \in \mathbb{R} : C(c_{\mathrm{tot}}, \mathrm{MAMU}_m) = O(m^\tau)\},$$

can be defined via the multiplicative complexity,

$$\omega = \omega_* = \omega_*(\mathbf{MAMU}) = \inf\{\tau \in \mathbb{R} : C(c_*, \mathrm{MAMU}_m) = O(m^\tau)\}$$

since ω_* can be defined by restricting to bilinear algorithms ([St2]). (This is the motivation for counting only non-linear steps. For the latest bounds on ω we refer to [CW], [St3]).

2. The non-uniform (semi-algebraic) complexity and the uniform complexity notion of Blum, Shub, and Smale [BSS] meet asymptotically in this case of matrix multiplication: if

$$\begin{aligned} \omega_{\mathrm{BSS}}(\mathbf{MAMU}) = \inf\{\tau : \exists \text{ BSS algorithm } \mathcal{A} \text{ for } \mathbf{MAMU} \text{ with time} \\ \text{bound } C(c_{\mathrm{tot}}, \mathcal{A}|\mathrm{MAMU}_m) = O(m^\tau) \text{ for } m \to \infty\}, \end{aligned}$$

then $\omega_{\mathrm{BSS}}(\mathbf{MAMU}) = \omega$.

Remark 1. Furthermore by Schönhage ([Schö2], p. 67), there is even an algorithm $\mathcal{A}$ for **MAMU** over $\mathbb{Q}$ as coefficient field with a meaningful variant in the bit-model satisfying for every $\tau > \omega$

$$C(c_{\mathrm{tot}}, \mathcal{A}|\mathrm{MAMU}_m) = o(m^\tau) \text{ as } m \to \infty.$$

$\mathcal{A}$ is designed in a multi-tasking "researching on all fronts" fashion to find better and better algorithms for the various MAMU_m.

■

For other sequences of $m \times m$ matrix computational tasks $\mathcal{S} = (\mathcal{S}_m : m \in \mathrm{N}_+)$, where such an asymptotic uniformization does not necessarily exists, we define its asymptotic exponent as

$$\omega_{\mathrm{BSS}}(\mathcal{S}) = \inf\{\tau : \exists \text{ BSS algorithm } \mathcal{A} \text{ for } \mathcal{S} \text{ with time bound } C(c_{\mathrm{tot}}, \mathcal{A}|\mathcal{S}_m) = O(m^\tau) \text{ for } m \to \infty\}.$$

In [Li1] it is shown that the asymptotic exponent of matrix multiplication with respect to additive complexity also coincides with ω,

$$\omega_+(\mathbf{MAMU}) = \omega_*(\mathbf{MAMU}) = \omega.$$

We are going to prove a similar complexity relativization for the additive complexity of the orthogonal basis computation task and the diagonalization task. As above, a quadratic form is thought to be given as a symmetric matrix; its discriminant is then the determinant of this matrix. For the subset of symmetric $m \times m$ matrices we shall simply write $R^{m(m+1)/2}$.

For the rest of this paragraph we fix the matrix size $m \times m$ and drop the index m indicating the size; the next paragraph 4 complements the semi-algebraic lower bound by an uniform upper bound.

Theorem 2. *1. Let $\mathcal{T}$ be a semi-algebraic computation tree (signature (Ω^R, P) as above) of input-output format (m^2, m^2) for the orthogonal basis computation task* OGB. *Then for every minimal prime cone $\alpha \in \mathrm{Min}(\widetilde{R^{m(m+1)/2}})$ having a specialization $\beta \in \mathrm{Min}(\widetilde{Z(\mathrm{dis})})$ the additive path length is bounded below as*

$$L(c_+, \mathcal{T}_\alpha) \geq \text{ const. } C(c_*, \mathrm{MAMU}).$$

2. Let $\mathcal{T}$ be a semi-algebraic computation tree of input-output format (m^2, m^2+m^2) for the diagonalization task DIAG. *Then for every minimal prime cone $\alpha \in \widetilde{\mathrm{Min}(R^{m(m+1)/2})}$ the additive path length is bounded below as*

$$L(c_+, \mathcal{T}_\alpha) \geq \textit{ const. } C(c_*, \mathrm{MAMU}).$$

Remark 2. We comment on the geometric signification of the two lower bound statements.

The first statement of the theorem implies for given $\mathcal{T}$ the existence of an open semi-algebraic $U \subset R^{m(m+1)/2}$, the intersection of which with $\mathcal{Z}(\mathrm{dis})$ is a non-void Zariski open subset of $\mathcal{Z}(\mathrm{dis})$, such that

$$L(c_+, \mathcal{T}_S) \geq \text{ const. } C(c_*, \mathrm{MAMU})$$

for every symmetric matrix $S \in U \setminus \mathcal{Z}(\mathrm{dis})$. Moreover, as the discriminant ideal $(\mathrm{dis}) \in \mathrm{Spec}\,\mathcal{P}(R^{m(m+1)/2})$ is central (cf. [BCR]), every matrix $S \in \mathcal{Z}(\mathrm{dis})$ appears in the closure of $U \setminus \mathcal{Z}(\mathrm{dis})$.

The second statement of the theorem implies for given $\mathcal{T}$ the lower bound

$$L(c_+, \mathcal{T}_S) \geq \text{ const. } C(c_*, \mathrm{MAMU})$$

for every symmetric matrix S in some non-void Zariski open $U \subset R^{m(m+1)/2}$.

■

Proof. 1. The organization of this proof is divided into showing a relative lower bound and an absolute one. First we show for a small and a big constant

$$L(c_+, \mathcal{T}_\alpha) \geq (\text{small}) \cdot C(c_*, \mathrm{MAMU}_m) - (\text{big}) \cdot m^2 \geq (\text{small}) \cdot m^\omega - (\text{big}) \cdot m^2. \tag{4}$$

In order to get rid of the "ω-ignorabimus" we then show

$$L(c_+, \mathcal{T}_{\alpha,\beta}) \geq \frac{(m-1)(m-2)}{2} \tag{5}$$

for the common path $\mathcal{T}_{\alpha,\beta}$ of α and its specialization $\beta \in \widetilde{\mathrm{Min}(\mathcal{Z}(\mathrm{dis}))}$ which is a initial segment of $\mathcal{T}_\alpha$; here the specialization β is needed.

Let $\mathcal{T}$ be a corresponding computation tree. Let

$$S \in R[S_{ij} : 1 \leq i \leq j \leq m]^{m \times m}$$

be the symmetrix matrix with entry S_{ij} for $1 \leq i \leq j \leq m$, and let $M \in R(S)^{m \times m}$ be that matrix such that $M(\alpha)$ is the output $\mathcal{T}(\alpha)$ of $\mathcal{T}$ on input $S(\alpha)$. Passing from OGB to its tilda $\widetilde{\text{OGB}}$ we conclude

$${}^tM(\alpha)S(\alpha)M(\alpha) = D(\alpha) = \begin{pmatrix} d_1(\alpha) & & \\ & \ddots & \\ & & d_m(\alpha) \end{pmatrix}$$

for some regular diagonal matrix $D \in R(S)^{m \times m}$.

Using three matrix times vector multiplications we compute with $3(m-1)m$ additions from $S(o)$, M, and the vector ${}^t(1, \ldots, 1)$ the diagonal elements $d_1, \ldots, d_m \in K = R(S)$, and then with further $(m-1)m$ additions (to compute the trace of the product of two matrices)

$$\operatorname{trace} S(o)^{-1} = \operatorname{trace}(M D^{-1})\, {}^tM.$$

Altogether we have so far (similarly as in [BKL]),

$$L(c_+, S(o), \operatorname{trace} S(o)^{-1}) \leq L(c_+, \mathcal{T}_\alpha) + 4(m-1)m. \tag{6}$$

In the further demonstration we do no longer protocoll the constants which in principle are given explicitely.

Now Theorem 1 takes action. The expansion

$$(S(o) + Y)^{-1} = S(o)^{-1} - S(o)^{-1}YS(o)^{-1} + \cdots \in K[[Y]]^{m \times m},$$

equation (6), and Theorem 1 used for $d = 3$ imply (using further K-linear operation) for suitable constants

$$L(c_*, Y, \operatorname{trace} S(o)^{-1}YS(o)^{-1}YS(o)^{-1}YS(o)^{-1}) \leq \text{const. } L(c_+, \mathcal{T}_\alpha) + \text{Const. } m^2; \tag{7}$$

the removal of

$$\operatorname{trace}(S(o)^{-1} - S(o)^{-1}YS(o)^{-1} + (S(o)^{-1}Y) \cdot (S(o)^{-1}YS(o)^{-1}))$$

can be arranged with additional m^2 many K-nonlinear multiplications by an extra computation of it since computation of the trace of a product of two matrices from these requires at most m^2 multiplications.

For square matrices $\mathfrak{A}, \mathfrak{B}$ we use now the abbreviation

$$\tau(\mathfrak{A}, \mathfrak{B}) = \text{trace}\, \mathfrak{A}\mathfrak{B}\mathfrak{A}\mathfrak{B}\mathfrak{A}\mathfrak{B}\mathfrak{A}\mathfrak{B}\mathfrak{A}.$$

Within the subsequent reasonings we shall apply several morphisms in **flac**. (That is to say, the variable standpoint with respect to coefficients – rather than the traditional fixed coefficient field view – will be of particular importance in our argumentation.)

We choose a prime cone $\gamma \in \text{Spec}_r K$ (possibly different from the α above) such that the matrix

$$S(\gamma)^{-1} \in k(\gamma)^{m \times m}$$

becomes positive definite, an orthogonal matrix $U \in \mathbf{O}_{k(\gamma)}(m)$ (classical orthogonal group) and elements $e_1, \ldots, e_m \in k(\gamma)$ such that

$$US(\gamma)^{-1}U^{-1} = E^2 = \begin{pmatrix} e_1^2 & & \\ & \ddots & \\ & & e_m^2 \end{pmatrix}$$

is diagonal. First of all, the assignment (in matrix notation)

$$S(o)^{-1} \mapsto 1 + S(o)^{-1}$$

defines an element in $\text{Aut}_R K$ the induced coefficient conjugation of which on $K[Y]$ (an isomorphism in **flac**) implies by (7)

$$L(c_*, Y, \tau(1 + S(o)^{-1}, Y)) \leq \text{const.}\ L(c_+, \mathcal{T}_\alpha) + \text{Const.}\ m^2 \qquad (8)$$

with the same constants as in (7). Performing scalar extension with $k(\gamma)$, (7) implies also

$$L(c_*, Y^{k(\gamma)}, \tau(U^{-1}E^2U, Y^{k(\gamma)})) \leq \text{const.}\ L(c_+, \mathcal{T}_\alpha) + \text{Const.}\ m^2, \qquad (9)$$

and in addition, (8) yields

$$L(c_*, Y^{k(\gamma)}, \tau(U^{-1}(1 + E^2)U, Y^{k(\gamma)})) \leq \text{const.}\ L(c_+, \mathcal{T}_\alpha) + \text{Const.}\ m^2. \qquad (10)$$

Now we inspect the slightly "baroque" expression

$$\begin{aligned} &\tau(U^{-1}E^2U, Y^{k(\gamma)}) \\ &= \text{trace}\, U^{-1}E(EUYU^{-1}E)(EUYU^{-1}E)(EUYU^{-1}E)EU \end{aligned}$$

and the corresponding one with E replaced by $\sqrt{1+E^2}$ (positive roots of all diagonal elements) for $\tau(U^{-1}(1+E^2)U, Y^{k(\gamma)})$ to observe that with the help of the $k(\gamma)$-algebra morphism of substitution (written in matrix notation)

$$Y \mapsto U^{-1}E^{-1}YE^{-1}U$$

inequality (9) and inequality (10), with the help of the substitution

$$Y \mapsto U^{-1}(\sqrt{1+E^2})^{-1}Y(\sqrt{1+E^2})^{-1}U,$$

imply the complexity bounds

$$L(c_*, Y^{k(\gamma)}, \operatorname{trace} E(Y^{k(\gamma)})^3E) \leq \text{const. } L(c_+, \mathcal{T}_\alpha) + \text{Const. } m^2, \quad (11)$$

and

$$\begin{aligned} L(c_*, Y^{k(\gamma)}, \operatorname{trace} \sqrt{1+E^2}\,(Y^{k(\gamma)})^3\,\sqrt{1+E^2}) \leq \\ \text{const. } L(c_+, \mathcal{T}_\alpha) + \text{Const. } m^2 \end{aligned} \quad (12)$$

since $k(\gamma)$-linear operations are not counted. (Note that counting *only* non-linear operations is an essential point to make the whole proof arrangement possible).

Next we use the straight line program transformation for the gradient of [Lin] and [BS]. We first note the following trivial fact.

Lemma 2 *For a ring A and the polynomial*

$$t = \operatorname{trace} X \cdot Y \cdot Z = \sum_{i,j,k=1}^{m} X_{ij}Y_{jk}Z_{ki} \in A[X, Y, Z]$$

one has for the partials

$$\frac{\partial t}{\partial X_{ij}} = (Y \cdot Z)_{ji}, \quad \frac{\partial t}{\partial Y_{jk}} = (Z \cdot X)_{kj}, \quad \frac{\partial t}{\partial Z_{ki}} = (X \cdot Y)_{ik}.$$

Proof. Since t is invariant under cyclic permutation of the three matrices the last evident equality implies the first two.

Now we differentiate the traces in (11) and (12). Differentiation and a correction of some scalar factors of 2 show for the matrices

$$V = (Y^{k(\gamma)})^2E^2 + Y^{k(\gamma)}E^2Y^{k(\gamma)} + E^2(Y^{k(\gamma)})^2$$

and

$$W = (Y^{k(\gamma)})^2(1+E^2) + Y^{k(\gamma)}(1+E^2)Y^{k(\gamma)} + (1+E^2)(Y^{k(\gamma)})^2$$

the complexity bounds

$$L(c_*, Y^{k(\gamma)}, V) \leq \text{const. } L(c_+, \mathcal{T}_\alpha) + \text{Const. } m^2,$$

$$L(c_*, Y^{k(\gamma)}, W) \leq \text{const. } L(c_+, \mathcal{T}_\alpha) + \text{Const. } m^2.$$

Since $W - V = 3 \cdot (Y^{k(\gamma)})^2$ we can conclude

$$L(c_*, Y^{k(\gamma)}, (Y^{k(\gamma)})^2) \leq \text{const. } L(c_+, \mathcal{T}_\alpha) + \text{Const. } m^2.$$

Let $m = 4l + \delta$ for some $\delta \leq 3$. By a substitution of the $m \times \delta$ east block and the $\delta \times m$ south block to zero we can reduce the matrix size to size $4l \times 4l$. So we assume $m = 4l$ in what follows. The $k(\gamma)$-algebra morphism of substitution canonically induced by the R-algebra substitution $R[Y] \to R[A, B]$, given in matrix notation by

$$Y \mapsto \begin{pmatrix} 0 & 0 & {}^tA & 0 \\ 0 & 0 & {}^tB & 0 \\ A & B & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

shows for $l \times l$ matrix multiplication the complexity bound

$$\begin{aligned} L(c_*, AB, {}^tA \cdot B) &= L(c_*, A^{k(\gamma)} B^{k(\gamma)}, ({}^tA \cdot B)^{k(\gamma)}) \\ &\leq \text{const. } L(c_+, \mathcal{T}_\alpha) + \text{Const. } m^2, \end{aligned}$$

by transfering back the complexity bound over $k(\gamma)$ to coefficient field R (see the remark below); here, as in [Li2], we use the notation $AB \in \boldsymbol{A}^{l \times l + l \times l}_{R \to R[A,B]}$ for the concatenated list $AB = (A_{11}, \ldots, A_{ll}; B_{11}, \ldots, B_{ll})$. Using block matrix calculation in order to get back to $m \times m$ matrix size, this completes the proof of the bound (4).

Remark 3. Let $\text{pat} = (r, c, c', a, s, m; n, l, d) \in \mathrm{N}^6 \times \mathrm{N}^3$ be a pattern of numbers. Then there is a first order formula $\Phi_{\text{pat}}(\text{coeff})$ of the language of ordered fields with parameters in $\mathbb{Z}$ – the package of free variables of which is denoted coeff – such that for every real closed field R and every list of polynomials $f \in \boldsymbol{A}^l_{R \to R[X_1, \ldots, X_n]}$, each f_i having degree at most d, $\Phi_{\text{pat}}(\text{coefficient system of } f)$ is true in R if and only if there exists a

division free Ω^R-SLP over n computing f from X with r many R-scalar multiplication, c nullary constant 0, c' nullary constant 1, a addition, s subtraction, and m multiplication instructions.

As a consequence (by the Tarski-Seidenberg principle), for an extension $\mathcal{R} \supset R$ of real closed fields and every cost function $c : \Omega^R \to \mathrm{N}$ constant on R-scalar multiplications and division,

$$L(c, X, f) = L(c^{\mathcal{R}}, X^{\mathcal{R}}, f^{\mathcal{R}})$$

for every $f \in \mathbf{A}^1_{R \to R[X_1,\ldots,X_n]}$; here $c^{\mathcal{R}} : \Omega^{\mathcal{R}} \to \mathrm{N}$ denotes the extension of c being constant on $\mathcal{R}$-scalar multiplications. In other words, the scalar extension with $\mathcal{R}$ is an autarkical monomorphism in flac with respect to the complexity data (c, X) and $(c^{\mathcal{R}}, X^{\mathcal{R}})$ (cf. [Li2]).

We are now going to prove the absolute lower bound (5) on the common path $\mathcal{T}_{\alpha,\beta}$ of both, α and β. According to whether the paths $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$ split or not we distinguish two cases:

Case $\mathcal{T}_\alpha \neq \mathcal{T}_\beta$: Since $\mathcal{T}$ distinguishes α and its specialization β the last comparision in the path $\mathcal{T}_{\alpha,\beta}$ yields an isolation of $\beta \in \mathrm{Spec}_r R[S]_{\mathrm{supp}\,\beta}$ (see [Li2]). (Note that $\mathcal{T}_{\alpha,\beta}$ does not necessarily verify β; if necessary, one first has to replace the last comparison by an equality test.) Hence by Proposition 1 and Lemma 1

$$L(c_+, \mathcal{T}_{\alpha,\beta}) \geq \frac{m(m-1)}{2} - 1,$$

and the bound (5) is guaranteed.

Case $\mathcal{T}_\alpha = \mathcal{T}_\beta$: If there is no split of paths, let $\mathfrak{p} = (\mathrm{dis}) \in R[S_{ij} : 1 \leq i \leq j \leq m]$, $S \in R[S_{ij} : 1 \leq i \leq j \leq m]^{m \times m}$ be the symmetric matrix with entry S_{ij} for $1 \leq i \leq j \leq m$, and let $M \in R[S]^{m \times m}_{\mathfrak{p}}$ be that regular matrix such that $M(\alpha)$ and $M(\beta)$ are the outputs $\mathcal{T}(\alpha)$ resp. $\mathcal{T}(\beta)$ of $\mathcal{T}$ on input $S(\alpha)$ resp. $S(\beta)$. Considering again the tilda OGB we conclude for $\gamma \in \{\alpha, \beta\}$

$${}^tM(\gamma)S(\gamma)M(\gamma) = D(\gamma) = \begin{pmatrix} d_1(\gamma) & & \\ & \ddots & \\ & & d_m(\gamma) \end{pmatrix}$$

for some diagonal matrix $D \in R[S]^{m \times m}_{\mathfrak{p}}$ such that $D(\alpha)$ is regular. Since $M(\beta)$ is regular and $S(\beta)$ has rank $m - 1$, exactly for one diagonal

element, say the first one, $d_1(\beta) = 0$, but $d_1(\alpha) \neq 0$. In order to get an isolation for $\beta \in \mathrm{Spec}_r R[S]_{\mathfrak{p}}$ we multiply the above matrix equation from the right with the inverse $M^{-1}(\gamma)$ – which we do not compute! – getting

$$^tM(\gamma)S(\gamma) = D(\gamma)M^{-1}(\gamma).$$

Since $M(\beta)$ is regular at least one element in the first row of $M^{-1}(\beta)$ must be non-zero, say $(M^{-1})_{1j}(\beta) \neq 0$. Considering the left side of of this matrix equation we see that one can compute the element $d_1 \cdot (M^{-1})_{1j} \in R[S]_{\mathfrak{p}}$ from M and $S_{\mathfrak{p}}$ with additional $m-1$ additions, and this element provides an isolation of $\beta \in \mathrm{Spec}_r R[S]_{\mathfrak{p}}$. Hence by Proposition 1 and Lemma 1

$$L(c_+, \mathcal{T}_{\alpha,\beta}) \geq \frac{\overline{m(m-1)}}{2} - m + 1,$$

and the bound (5) is guaranteed in this case as well.

This completes the proof of the first part of Theorem 2.

2. The proof of the second part is similar to the first one. The relative lower bound proof above did not require α to possess a specialization in $\mathrm{Min}(\widetilde{\mathcal{Z}(\mathrm{dis})})$, and remains valid here. For the absolute lower bound we make a slightly different distinction. Let again $\mathfrak{p} = (\mathrm{dis}) \in R[S_{ij} : 1 \leq i \leq j \leq m]$. If the SLP $\Gamma(\mathcal{T}_\alpha)$ of purely computational steps along $\mathcal{T}_\alpha$ is not executable on $S(\mathfrak{p})$ then its execution on $S_{\mathfrak{p}}$ produces a non-zero non-unit in $R[S]_{\mathfrak{p}}$, so an isolation of its maximal ideal, and

$$L(c_+, \mathcal{T}_\alpha) \geq \frac{m(m-1)}{2}. \tag{13}$$

If it is executable, let $M, D \in R[S]_{\mathfrak{p}}^{m \times m}$ be those regular matrices such that the pair $(M(\alpha), D(\alpha))$ is the output $\mathcal{T}(\alpha)$ of $\mathcal{T}$ on input $S(\alpha)$. Evaluation in $\mathfrak{p}$,

$$^tM(\mathfrak{p})S(\mathfrak{p})M(\mathfrak{p}) = D(\mathfrak{p}),$$

shows that one of the diagonal elements of D provides an isolation of $\mathfrak{p} \in \mathrm{Spec}\, R[S]_{\mathfrak{p}}$. So inequality (13) in valid in this case too.

■

4 BSS Discussion

The lower bounds of the last paragraph can be complemented by uniform upper bounds, even for total complexity.

Theorem 3. *Asymptotically, for $m \to \infty$, diagonalization, orthogonal basis computation, and matrix multiplication have the same total complexity,*

$$\omega_{\mathrm{BSS}}(\mathbf{DIAG}) = \omega_{\mathrm{BSS}}(\mathbf{OGB}) = \omega.$$

Proof. We describe a BSS algorithm $\mathcal{A}$ for **DIAG** assuming to have given for $\tau > 2$ an algorithm for **MAMU** with time bound $O(m^\tau)$ when restricted on MAMU_m. $\mathcal{A}$ is calling three main $O(m^\tau)$ subroutines to be described first.

- $\mathcal{A}_{\mathrm{reg}}$ (regularization): This algorithm receives a symmetric $m \times m$ matrix S, finds its rank $r \leq m$, and computes a regular $m \times m$ matrix M and a regular symmetric $r \times r$ matrix R such that

$$^tMSM = \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix}.$$

Via fast matrix multiplication this can be done with Keller-Gehrig's [Ke] elimination variant of Schönhage's [Schö1] triangulation which computes ^tM and the product ^tMS such that this product is in eliminated staircase form.

- $\mathcal{A}_{\mathrm{split},l}$ (direct sum decomposition): This algorithm receives a symmetric $m \times m$ matrix S with a regular symmetric $l \times l$ north-west block R and computes a regular $m \times m$ matrix M and the product ^tMSM such that the latter is of the form

$$^tMSM = \begin{pmatrix} R & 0 \\ 0 & T \end{pmatrix}$$

with the same regular symmetric $l \times l$ north-west block R.

A block elimination can be arranged using fast inversion ([St1], [Schö1], [Schö3], [Ke]) in the format $l \times l$ and $m \times m$ matrix multiplication yielding

$$^tM = \begin{pmatrix} 1 & 0 \\ -AR^{-1} & 1 \end{pmatrix} \quad \text{if} \cdot S = \begin{pmatrix} R & {}^tA \\ A & U \end{pmatrix}.$$

- $\mathcal{A}_{\mathrm{nw-reg}}$ (north-west-regularization): This algorithm receives a regular symmetric $m \times m$ matrix S, $m = 2l + \delta$ for some $\delta \in \{0, 1\}$, and computes a regular $m \times m$ matrix M and ${}^tMSM$ such that this product is of the form

$$ {}^tMSM = \begin{pmatrix} R & {}^tA \\ A & T \end{pmatrix} $$

with regular symmetric $l \times l$ north-west block R.

First we apply $\mathcal{A}_{\mathrm{reg}}$ to the $l \times l$ north-west block of S, find its rank $r \leq l$, and assume for the further description the $l \times l$ north-west block of S already to be in the form

$$ \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} $$

with regular symmetric $r \times r$ matrix R. Then an application of $\mathcal{A}_{\mathrm{split},r}$ in the format $(r + l + \delta) \times (r + l + \delta)$ and a matrix size adjustment manufactures a regular $m \times m$ matrix M such that the product ${}^tMSM$ is of the form

$$ {}^tMSM = \begin{pmatrix} R & 0 & 0 \\ 0 & 0 & {}^tA \\ 0 & A & T \end{pmatrix} $$

where the $(l + \delta) \times (l - r)$ block A has rank $l - r$ since S is regular. A transformation of the whole $(l + \delta) \times m$ south block into eliminated staircase form yields a further regular $m \times m$ matrix M' such that the product ${}^tM' {}^tMSMM'$ is of the form

$$ {}^tM' {}^tMSMM' = \begin{pmatrix} R & 0 & 0 & 0 \\ 0 & 0 & {}^tB & 0 \\ 0 & B & U & {}^tC \\ 0 & 0 & C & V \end{pmatrix}. $$

This whole matrix is transformed into a form with regular $l \times l$ north-west block once the inner north-west null and south-west regular $2(l-r) \times 2(l-r)$ block is transformed into north-west regular form; by a $2(l-r) \times 2(l-r)$ block permutation matrix P we transform this block into the form

$$ {}^tP \begin{pmatrix} 0 & {}^tB \\ B & U \end{pmatrix} P = \begin{pmatrix} U & {}^tB \\ B & 0 \end{pmatrix}. $$

The goal is reached if U is regular, or in case of a hyperbolic shape $U = 0$ and $B = 1$ after performing a simple transformation. The general case is reduced to these two cases as follows. We apply again $\mathcal{A}_{\mathrm{reg}}$ to the north-west block U, find its rank $s \leq l-r$, transform the whole $2(l-r)\times 2(l-r)$ matrix accordingly, invert then the resulting south-west block to obtain finally a transformed shape

$$\begin{pmatrix} W & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

with regular symmetric $s \times s$ matrix W. Transformation of the hyperbolic block achieves the goal. Multiplying all partial transformation matrices together we obtain the overall transformation matrix.

The algorithm $\mathcal{A}$ now works as follows. Given a symmetric $m \times m$ matrix S, $\mathcal{A}$ first calls $\mathcal{A}_{\mathrm{reg}}$, finds its rank $r \leq m$, and computes a regular $m \times m$ matrix M and a regular symmetric $r \times r$ matrix R such that

$${}^tMSM = \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix}.$$

Then $\mathcal{A}$ calls $\mathcal{A}_{\mathrm{nw-reg}}$ for the regular symmetric $r\times r$ matrix R and manufactures a regular $m\times m$ matrix M' such that the product ${}^tM'\,{}^tMSMM'$ has a regular $\lfloor r/2\rfloor \times \lfloor r/2\rfloor$ north-west block. Then $\mathcal{A}$ calls $\mathcal{A}_{\mathrm{split},\lfloor r/2\rfloor}$ and manufactures a regular $m \times m$ matrix M'' such that

$${}^tM''\,{}^tM'\,{}^tMSMM'M'' = \begin{pmatrix} T & 0 & 0 \\ 0 & U & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

is in block diagonal form with regular symmetric $\lfloor r/2\rfloor \times \lfloor r/2\rfloor$ matrix T and regular symmetric $\lceil r/2\rceil \times \lceil r/2\rceil$ matrix U; if $r \notin 2\mathbb{Z}$ then an additional traditional one column elimination is used to have U in the form

$$U = \begin{pmatrix} V & 0 \\ 0 & W \end{pmatrix}$$

with regular symmetric $\lfloor r/2\rfloor \times \lfloor r/2\rfloor$ matrix V. Then $\mathcal{A}$ is recursively called twice for arguments T and V. Although matrix multiplication and related algorithms being called many times, $\tau > 2$ and a geometric

series argument as described above for matrix multiplication lead to the upper bound

$$C(c_{\text{tot}}, \mathcal{A}|\text{DIAG}_m) \leq 2 \cdot C(c_{\text{tot}}, \mathcal{A}|\text{DIAG}_{\lfloor r/2 \rfloor}) + \text{ const. } m^\tau = O(m^\tau).$$

As $\tau > \omega \geq 2$ was arbitrary this shows $\omega_{\text{BSS}}(\mathbf{OGB}) \leq \omega_{\text{BSS}}(\mathbf{DIAG}) \leq \omega$.

■

Acknowledgments We thank the referees for their useful remarks.

References

[BS] W. Baur and V. Strassen, *The complexity of partial derivatives*, Theoret. Comput. Sci. **22** (1983), 317–330.

[Be] E. Becker, *On the real spectrum of a ring and its applications to semi-algebraic geometry*, Bull. Am. Math. Soc. (N. S.) **15** (1986), 19–60.

[BSS] L. Blum, M. Shub, and S. Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions, and universal machines*, Bull. Amer. Math. Soc. **21** (1989), 1–46.

[BKL] P. Bürgisser, M. Karpinski, and T. Lickteig, *Some computational problems in linear algebra as hard as matrix multiplication*, Comput. Complexity 1 (1991), 131–155.

[BCR] J. Bochnak, M. Coste, and M.-F. Roy, *Géométrie algébrique réelle*, Ergebnisse der Mathematik und ihrer Grenzgebiete, 3. Folge, Band **12**, Springer-Verlag, 1987.

[CW] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, J. Symb. Comput. **9** (1990), 251–280.

[CR] M. Coste and M.-F. Roy, *La Topologie du Spectre Réel*, Contemporary Mathematics, Vol. 8 (1992), 27–59.

[dG] H.-F. de Groote, *Lectures on the complexity of bilinear problems*, Lecture Notes in Computer Science 245, Springer-Verlag, 1987.

[GHMP] M. Giusti, J. Heintz, J. E. Morais, and L. M. Pardo, *When polynomial equation systems can be "solved" fast?, in: Applied Algebra, Algebraic Algorithms and Error-Correcting Codes,* pp. 205–231, Lecture Notes in Computer Science 948, Springer-Verlag, 1995.

[Gr] D. Yu. Grigoryev, Notes of scientific seminars of LOMI, Vol. 118, Sankt Petersburg (1982), 25–82.

[HM] J. Heintz and J. Morgenstern, *On the Intrinsic Complexity of Elimination Theory,* J. Complexity **9**, 471–498 (1993).

[Ka] K. Kalorkoti, *The trace invariant and matrix multiplication,* Theoret. Comp. sci. **59** (1988), 277–286.

[Ke] W. Keller-Gehrig, *Fast algorithms for the characteristic polynomial,* Theoret. Comp. sci. **36** (1985), 309–317.

[Kh] A. G. Khovanskii, *On a class of systems of transcendental equations,* Soviet Math. Dokl. **22(3)** (1980), 762–765.

[KS] M. Knebusch and C. Scheiderer, *Einführung in die reelle Algebra,* Vieweg-Studium 63: Aufbaukurs Mathematik, Vieweg, 1989.

[Li1] T. Lickteig, *On semi-algebraic decision complexity,* Habilitationsschrift, Univ. Tübingen, and Tech. Rep. TR-90-052 Int. Comp. Sci. Inst., Berkeley, 1990.

[Li2] T. Lickteig, *Semi-algebraic Decision Complexity, the Real Spectrum, and Degree,* Journal of Pure and Applied Algebra **110(2)** (1996), 131–184.

[LM] T. Lickteig, K. Meer, *Semi-algebraic complexity – Additive complexity of matrix computational tasks,* Festschrift on occasion of Shmuel Winograd's 60th birthday, Journal of Complexity, **13**, to appear (1997).

[LR] T. Lickteig and M.-F. Roy, *Semi-algebraic complexity of quotients and sign determination of remainders,* Journal of Complexity, **12**, 545-571 (1996).

[Lin] S. Linnainmaa, *Taylor expansion of the accumulated rounding error,* BIT **16** (1976), 146–160.

[Mo] J. Morgenstern, *Algorithmes linéaires tangents et complexité,* C. R. Acad. Sci. Paris, t. **277** (1973), 367.

[vNG] J. von Neumann and H. H. Goldstine, *Numerical inverting of matrices of high order,* Bull. Amer. Math. Soc. **53** (1947), 1021–1099.

[Pa] V. Ya. Pan, *How to Multipy Matrices Faster,* Lecture Notes in Computer Science 179, Springer-Verlag, 1984.

[Ri] J. J. Risler, *Additive complexity and zeros of real polynomials,* SIAM J. Comput. **14** (1985), 178–183.

[Ro] M.-F. Roy, *Faisceau structural sur le spectre réel et fonctions de Nash, in: Géométrie algébrique réelle et formes quadratiques,* 406–432, Lecture Notes in Mathematics 959, Springer-Verlag, 1982.

[Schö1] A. Schönhage, *Unitäre Transformation großer Matrizen,* Numerische Mathematik **20** (1973), 409–417.

[Schö2] A. Schönhage, *The fundamental theorem of algebra in terms of computational complexity,* Technical Report, Univ. Tübingen, 1982, 74 pp.

[Schö3] A. Schönhage, *Equation solving in terms of computational complexity,* in: Proc. Internat. Congress of Mathematicians 1986, Berkeley, 131–153.

[St1] V. Strassen, *Gaussian elimination is not optimal,* Num. Mathematik **13** (1969), 354–356.

[St2] V. Strassen, *Vermeidung von Divisionen,* J. Reine Angew. Math. **264** (1973), 184–202.

[St3] V. Strassen, *Relative bilinear complexity and matrix multiplication,* J. Reine Angew. Math. **375/376** (1987), 406–443.

[Tu] A. M. Turing, *Rounding-off errors in matrix processes,* Quart. J. Mech. **1** (1948), 287–308.

[Wi] S. Winograd, *On the number of multiplications necessary to compute certain functions,* Comm. Pure Appl. Math. **23** (1970), 165 – 179.

Institut für Informatik
Universität Bonn
Römerstr. 164
53117 Bonn
Germany
lickteig@zariski.cs.uni-bonn.de

RWTH Aachen
c/o Lehrstuhl C für Mathematik
Templergraben 55 52062 Aachen
Germany
meer@rwth-aachen.de