HOW TO INTEGRATE A POLYNOMIAL OVER A SIMPLEX

V. BALDONI, N. BERLINE, J. A. DE LOERA, M. KÖPPE, AND M. VERGNE

Abstract. This paper settles the computational complexity of the problem of integrating a polynomial function f over a rational simplex. We prove that the problem is NP-hard for arbitrary polynomials via a generalization of a theorem of Motzkin and Straus. On the other hand, if the polynomial depends only on a fixed number of variables, while its degree and the dimension of the simplex are allowed to vary, we prove that integration can be done in polynomial time. As a consequence, for polynomials of fixed total degree, there is a polynomial time algorithm as well. We conclude the article with extensions to other polytopes and discussion of other available methods.

1. Introduction

Let Δ be a d-dimensional rational simplex inside $\mathbb{R}^n$ and let $f \in \mathbb{Q}[x_1, \ldots, x_n]$ be a polynomial with rational coefficients. We consider the problem of how to efficiently compute the *exact* value of the integral of the polynomial f over Δ, which we denote by $\int_\Delta f \, \mathrm{d}m$. We use here the *integral Lebesgue measure* $\mathrm{d}m$ on the affine hull $\langle \Delta \rangle$ of the simplex Δ, defined below in section 2.1. This normalization of the measure occurs naturally in Euler–Maclaurin formulas for a polytope P, which relate sums over the lattice points of P with certain integrals over the various faces of P. For this measure, the volume of the simplex and every integral of a polynomial function with rational coefficients are *rational numbers*. Thus the result has a representation in the usual (Turing) model of computation. This is in contrast to other normalizations, such as the induced Euclidean measure, where irrational numbers appear.

The main goals of this article are to discuss the computational complexity of the problem and to provide methods to do the computation that are both theoretically efficient and have reasonable performance in concrete examples.

Computation of integrals of polynomials over polytopes is fundamental for many applications. We already mentioned summation over lattice points of a polytope. They also make an appearance in recent results in optimization problems connected to moment matrices [23]. These integrals are also commonly computed in finite element methods, where the domain is decomposed into cells (typically simplices) via a

mesh and complicated functions are approximated by polynomials (see for instance [31]). When studying a random univariate polynomial $p(x)$ whose coefficients are independent random variables in certain intervals, the probability distribution for the number of real zeros of $p(x)$ is given as an integral over a polytope [9]. Integrals over polytopes also play a very important role in statistics, see, for instance, [25]. Remark that among all polytopes, simplices are the fundamental case to consider for integration since any convex polytope can be triangulated into finitely many simplices.

Regarding the computational complexity of our problem, one can ask what happens with integration over arbitrary polytopes. It is very educational to look first at the case when f is the constant polynomial 1, and the answer is simply a volume. It has been proved that already computing the volume of polytopes of varying dimension is #P-hard [16, 12, 20, 24], and that even approximating the volume is hard [17]. More recently in [28] it was proved that computing the centroid of a polytope is #P-hard. In contrast, for a simplex, the volume is given by a determinant, which can be computed in polynomial time. One of the key contributions of this paper is to settle the computational complexity of integrating a non-constant polynomial over a simplex. Before we can state our results let us understand better the input and output of our computations. Our output will always be the rational number $\int_\Delta f \, \mathrm{d}m$ in the usual binary encoding. The d-dimensional input simplex will be represented by its vertices $\mathbf{s}_1, \ldots, \mathbf{s}_{d+1}$ (a V-representation) but note that, in the case of a simplex, one can go from its representation as a system of linear inequalities (an H-representation) to a V-representation in polynomial time, simply by computing the inverse of a matrix.

Thus the encoding size of Δ is given by the number of vertices, the dimension, and the largest binary encoding size of the coordinates among vertices. Computations with polynomials also require that one specifies concrete data structures for reading the input polynomial and to carry on the calculations. There are several possible choices. One common representation of a polynomial is as a sum of monomial terms with rational coefficients. Some authors assume the representation is *dense* (polynomials are given by a list of the coefficients of all monomials up to a given total degree r), while other authors assume it is *sparse* (polynomials are specified by a list of exponent vectors of monomials with non-zero coefficients, together with their coefficients). Another popular representation is by *straight-line programs*. A straight-line program which encodes a polynomial is, roughly speaking, a program without branches which enables us to evaluate it at any given point (see [14, 26] and references therein). As we explain in Section 2, general straight-line programs are *too compact* for our purposes, so instead we restrict to a subclass we call *single-intermediate-use (division-free) straight-line*

programs or *SIU straight-line programs* for short. The precise definition and explanation will appear in Section 2 but for now the reader should think that polynomials are represented as fully parenthesized arithmetic expressions involving binary operators $+$ and $\times$.

Now we are ready to state our first result.

Theorem 1 (Integrating general polynomials over a simplex is hard)**.** *The following problem is NP-hard. Input:*

- (I_1) *numbers $d, n \in \mathbb{N}$ in unary encoding,*
- (I_2) *affinely independent rational vectors $\mathbf{s}_1, \dots, \mathbf{s}_{d+1} \in \mathbb{Q}^n$ in binary encoding,*
- (I_3) *an SIU straight-line program Φ encoding a polynomial $f \in \mathbb{Q}[x_1, \dots, x_n]$ with rational coefficients.*

Output, in binary encoding:

- (O_1) *the rational number $\int_\Delta f \,\mathrm{d}m$, where $\Delta \subseteq \mathbb{R}^n$ is the simplex with vertices $\mathbf{s}_1, \dots, \mathbf{s}_{d+1}$ and $\mathrm{d}m$ is the integral Lebesgue measure of the rational affine subspace $\langle\Delta\rangle$.*

But we can also prove the following positive results.

Theorem 2 (Efficient integration of polynomials of fixed *effective* number of variables)**.** *For every fixed number $D \in \mathbb{N}$, there exists a polynomial-time algorithm for the following problem.*
Input:

- (I_1) *numbers $d, n, M \in \mathbb{N}$ in unary encoding,*
- (I_2) *affinely independent rational vectors $\mathbf{s}_1, \dots, \mathbf{s}_{d+1} \in \mathbb{Q}^n$ in binary encoding,*
- (I_3) *a polynomial $f \in \mathbb{Q}[X_1, \dots, X_D]$ represented by either an SIU straight-line program Φ of formal degree at most M, or a sparse or dense monomial representation of total degree at most M,*
- (I_4) *a rational matrix L with D rows and n columns in binary encoding, the rows of which define D linear forms $\mathbf{x} \mapsto \langle \ell_j, \mathbf{x}\rangle$ on $\mathbb{R}^n$.*

Output, in binary encoding:

- (O_1) *the rational number $\int_\Delta f(\langle \ell_1, \mathbf{x}\rangle, \dots, \langle \ell_D, \mathbf{x}\rangle) \,\mathrm{d}m$, where $\Delta \subseteq \mathbb{R}^n$ is the simplex with vertices $\mathbf{s}_1, \dots, \mathbf{s}_{d+1}$ and $\mathrm{d}m$ is the integral Lebesgue measure of the rational affine subspace $\langle\Delta\rangle$.*

In particular, the computation of the integral of a power of *one* linear form can be done by a polynomial time algorithm. This becomes false already if one considers powers of a quadratic form instead of powers of a linear form. Actually, we prove Theorem 1 by looking at powers Q^M of the Motzkin–Straus quadratic form of a graph.

Our method relies on properties of integrals of exponentials of linear forms. A. Barvinok had previously investigated these integrals and their computational complexity (see [3], [5]).

As we will see later, when its degree is fixed, a polynomial has a polynomial size representation in either the SIU straight-line program encoding or the sparse or dense monomial representation and one can switch between the three representations efficiently. The notion of formal degree of an SIU straight-line program will be defined in Section 2.

Corollary 3 (Efficient integration of polynomials of fixed degree)**.** *For every fixed number $M \in \mathbb{N}$, there exists a polynomial-time algorithm for the following problem. Input:*

- (I_1) *numbers $d, n \in \mathbb{N}$ in unary encoding,*
- (I_2) *affinely independent rational vectors $\mathbf{s}_1, \ldots, \mathbf{s}_{d+1} \in \mathbb{Q}^n$ in binary encoding,*
- (I_3) *a polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ represented by either an SIU straight-line program Φ of formal degree at most M, or a sparse or dense monomial representation of total degree at most M.*

Output, in binary encoding:

- (O_1) *the rational number $\int_\Delta f(\mathbf{x})\,\mathrm{d}m$, where $\Delta \subseteq \mathbb{R}^n$ is the simplex with vertices $\mathbf{s}_1, \ldots, \mathbf{s}_{d+1}$ and $\mathrm{d}m$ is the integral Lebesgue measure of the rational affine subspace $\langle\Delta\rangle$.*

Actually, we give two interesting methods that prove Corollary 3. First, we simply observe that a monomial with total degree M involves at most M variables. The other method is related to the polynomial Waring problem: we decompose a homogeneous polynomial of total degree M into a sum of M-th powers of linear forms.

In [22] Lasserre and Avrachenkov compute the integral $\int_\Delta f(\mathbf{x})\,\mathrm{d}m$ when f is a homogeneous polynomial, in terms of the corresponding polarized symmetric multilinear form (Proposition 18). We show that their formula also leads to a proof of Corollary 3. Furthermore, several other methods can be used for integration of polynomials of fixed degree. We discuss them in Section 4.

This paper is organized as follows: After some preparation in Section 2, the main theorems are proved in Section 3. In Section 4, we discuss extensions to other convex polytopes and give a survey of the complexity of other algorithms. Finally, in Section 5, we describe the implementation of the two methods of Section 3, and we report on a few computational experiments.

2. Preliminaries

In this section we prepare for the proofs of the main results.

2.1. Integral Lebesgue measure on a rational affine subspace of $\mathbb{R}^n$.

On $\mathbb{R}^n$ itself we consider the standard Lebesgue measure, which gives volume 1 to the fundamental domain of the lattice $\mathbb{Z}^n$. Let L be a rational linear subspace of dimension $d \le n$. We normalize the Lebesgue measure on L, so that the volume of the fundamental domain

TABLE 1. The representation of $(x_1^2 + \cdots + x_n^2)^k$ as a straight-line program

Intermediate	Comment
$q_1 = 0$	
$q_2 = x_1$	
$q_3 = q_2 \cdot q_2$	
$q_4 = q_1 + q_3$	Thus $q_4 = x_1^2$.
$q_5 = x_2$	
$q_6 = q_5 \cdot q_5$	
$q_7 = q_4 + q_6$	Now $q_7 = x_1^2 + x_2^2$.
$\vdots$	
$q_{3n-1} = x_n$	
$q_{3n} = q_{3n-1} \cdot q_{3n-1}$	
$q_{3n+1} = q_{3n-2} + q_{3n}$	Now $q_{3n+1} = x_1^2 + \cdots + x_n^2$.
$q_{3n+2} = 1$	
$q_{3n+3} = q_{3n+2} \cdot q_{3n+1}$	
$q_{3n+4} = q_{3n+3} \cdot q_{3n+1}$	
$\vdots$	
$q_{3n+k+2} = q_{3n+k+1} \cdot q_{3n+1}$	Final result.

of the intersected lattice $L \cap \mathbb{Z}^n$ is 1. Then for any affine subspace $L + \mathbf{a}$ parallel to L, we define the *integral Lebesgue measure* $\mathrm{d}m$ by translation. For example, the diagonal of the unit square has length 1 instead of $\sqrt{2}$.

2.2. **Encoding polynomials for integration.** We now explain our encoding of polynomials as SIU straight-line programs and justify our use of this encoding. We say that a polynomial f is represented as a (division-free) *straight-line program* Φ if there is a finite sequence of polynomial functions of $\mathbb{Q}[x_1, \ldots, x_n]$, namely $q_1, \ldots, q_k$, the so-called *intermediate results*, such that each q_i is either a variable $x_1, \ldots, x_n$, an element of $\mathbb{Q}$, or either the sum or the product of two preceding polynomials in the sequence and such that $q_k = f$. A straight-line program allows us to describe in polynomial space polynomials which otherwise would need to be described with exponentially many monomial terms. For example, think of of the representation of $(x_1^2 + \cdots + x_n^2)^k$ as monomials versus its description with only $3n+k+2$ intermediate results; see Table 2. The number of intermediate results of a straight line program is called its *length*. To keep track of constants we define the *size* of an intermediate result as one, unless the intermediate result is a constant in which case its size is the binary encoding size of the rational number. The *size* of a straight-line program is the sum of the sizes of the

intermediate results. The *formal degree* of an intermediate result q_i is defined recursively in the obvious way, namely as 0 if q_i is a constant of $\mathbb{Q}$, as 1 if q_i is a variable x_j, as the maximum of the formal degrees of the summands if q_i is a sum, and the sum of the formal degrees of the factors if q_i is a product. The *formal degree of the straight-line program* Φ is the formal degree of the final result q_k. Clearly the total degree of a polynomial is bounded by the formal degree of any straight-line program which represents it.

A favorite example to illustrate the benefits of a straight-line program encoding is that of the symbolic determinant of an $n \times n$ matrix. Its dense representation as monomials has size $\Theta(n!)$ but it can be computed in $O(n^3)$ operations by Gaussian elimination. See the book [14] as a reference for this concept.

From a monomial representation of a polynomial of degree M and n variables it is easy to encode it as a straight-line program: first, by going in increasing degree we can write a straight-line program that generates all monomials of degree at most M in n variables. Then for each of them compute the product of the monomial with its coefficient so the length doubles. Finally successively add each term. This gives a final length bounded above by four times the number of monomials of degree at most M in n variables.

Straight-line programs are quite natural in the context of integration. One would certainly not expand $(x_1^2 + \cdots + x_n^2)^k$ to carry on numeric integration when we can easily evaluate it as a function. More importantly, straight-line programs are suitable as an input and output encoding and data structure in certain symbolic algorithms for computations with polynomials, like factoring; see [14]. Since straight-line programs can be very compact, the algorithms can handle polynomials whose input and output encodings have an exponential size in a sparse monomial representation.

However, a problem with straight-line programs is that this input encoding can be so compact that the output of many computational questions cannot be written down efficiently in the usual *binary encoding*. For example, while one can encode the polynomial x^{2^k} with a straight-line program with only $k+1$ intermediate results (see Table 2), when we compute the value of x^{2^k} for $x = 2$, or the integral $\int_0^2 x^{2^k}\,\mathrm{d}x = 2^{2^k+1}/(2^k+1)$, the binary encoding of the output has a size of $\Theta(2^k)$. Thus the output, given in binary, turns out to be exponentially bigger than the input encoding. We remark that the same difficulty arises if we choose a sparse input encoding of the polynomial where not only the coefficients but also the exponent vectors are encoded in binary (rather than the usual unary encoding for the exponent vectors).

TABLE 2. The representation of a straight-line program for x^{2^k}, using iterated squaring

Intermediate	Comment
$q_1 = x$	
$q_2 = q_1 \cdot q_1$	
$q_3 = q_2 \cdot q_2$	
$\vdots$	
$q_{k+1} = q_k \cdot q_k$	Final result.

TABLE 3. The representation of a single-intermediate-use straight-line program for x^{2^k}; note that the iterated squaring method cannot be used

Intermediate	Comment
$q_1 = x$	
$q_2 = x$	
$q_3 = q_1 \cdot q_2$	Now $q_3 = x^2$, and q_1 and q_2 cannot be used anymore.
$q_4 = x$	
$q_5 = q_3 \cdot q_4$	Thus $q_5 = x^3$.
$\vdots$	
$q_{2^{k+1}-2} = x$	
$q_{2^{k+1}-1} = q_{2^{k+1}-3} \cdot q_{2^{k+1}-2}$	Final result.

This motivates the following variation of the notion of straight-line program: We say a (division-free) straight-line program is *single-intermediate-use*, or SIU for short, if every intermediate result is used only once in the definition of other intermediate results. (However, the variables $x_1, \ldots, x_n$ can be used arbitrarily often in the definition of intermediate results.) With this definition, all ways to encode the polynomial x^{2^k} require at least 2^k multiplications. An example SIU straight-line program is shown in Table 3. Clearly single-intermediate-use straight-line programs are equivalent, in terms of expressiveness and encoding complexity, to fully parenthesized arithmetic expressions using binary operators $+$ and $\times$.

2.3. **Efficient computation of truncated product of an arbitrary number of polynomials in a fixed number of variables.** The following result will be used in several situations.

Lemma 4. *For every fixed number $D \in \mathbb{N}$, there exists a polynomial time algorithm for the following problem.*

Input: a number M in unary encoding, a sequence of k polynomials $P_j \in \mathbb{Q}[X_1, \ldots, X_D]$ of total degree at most M, in dense monomial representation.
Output: the product $P_1 \cdots P_k$ truncated at degree M.

Proof. We start with the product of the first two polynomials. We compute the monomials of degree at most M in this product. This takes $O(M^{2D})$ elementary rational operations, and the maximum encoding length of any coefficient in the product is also polynomial in the input data length. Then we multiply this truncated product with the next polynomial, truncating at degree M, and so on. The total computation takes $O(kM^{2D})$ elementary rational operations. □

3. Proofs of the main results

Our aim is to perform an efficient computation of $\int_\Delta f \, \mathrm{d}m$ where Δ is a simplex and f a polynomial. We will prove first that this is not possible for f of varying degree under the assumption that P $\neq$ NP. More precisely, we prove that, under this assumption, an efficient computation of $\int_\Delta Q^M \, \mathrm{d}m$ is not possible, where Q is a quadratic form and M is allowed to vary.

In the next subsection we present an algorithm to efficiently compute the integral $\int_\Delta f \, \mathrm{d}m$ in some particular situations, most notably the case of arbitrary powers of linear forms.

3.1. Hardness for polynomials of non-fixed degree.

For the proof of Theorem 1 we need to extend the following well-known result of Motzkin and Straus [27]. In this section, we denote by Δ the $(n-1)$-dimensional canonical simplex $\{\, \mathbf{x} \in \mathbb{R}^n : x_i \geq 0, \ \sum_{i=1}^n x_i = 1 \,\}$, and we denote by $\mathrm{d}m$ the Lebesgue measure on the hyperplane $\{\, \mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1 \,\}$, normalized so that Δ has volume 1. For a function f on Δ, denote as usual $\|f\|_\infty = \max_{\mathbf{x}\in\Delta} |f(\mathbf{x})|$ and $\|f\|_p = (\int_\Delta |f|^p \, \mathrm{d}m)^{1/p}$, for $p \geq 1$. Recall that the *clique number* of a graph G is the largest number of vertices of a complete subgraph of G.

Theorem 5 (Motzkin–Straus)**.** *Let G be a graph with n vertices and clique number $\omega(G)$. Let $Q_G(\mathbf{x})$ be the Motzkin–Straus quadratic form $\frac{1}{2}\sum_{(i,j)\in E(G)} x_i x_j$. Then $\|Q_G(\mathbf{x})\|_\infty = \frac{1}{2}(1 - \frac{1}{\omega(G)})$.*

Our first result might be of independent interest as it shows that integrals of polynomials over simplices can carry very interesting combinatorial information. This result builds on the theorem of Motzkin and Straus, using the proof of the well-known relation $\|f\|_\infty = \lim_{p\to\infty} \|f\|_p$.

Lemma 6. *Let G be a graph with n vertices and clique number $\omega(G)$. Let $Q_G(\mathbf{x})$ be the Motzkin–Straus quadratic form. Then for $p \geq 4(e-1)n^3 \ln(32n^2)$, the clique number $\omega(G)$ is equal to $\left\lceil \frac{1}{1-2\|Q_G\|_p} \right\rceil$.*

To prove Lemma 6 we will first prove the following intermediate result.

Lemma 7. *For $\varepsilon > 0$ we have*

$$(\|Q_G\|_\infty - \varepsilon)(\frac{\varepsilon}{4})^{(n-1)/p} \leq \|Q_G\|_p \leq \|Q_G\|_\infty .$$

Proof. The right-hand side inequality follows from the normalization of the measure, as $|Q(\mathbf{x})| \leq \|Q_G\|_\infty$, for all $\mathbf{x} \in \Delta$.

In order to obtain the other inequality, we use Hölder's inequality $\int_\Delta |fg| \, \mathrm{d}m \leq \|f\|_p \|g\|_q$, where q is such that $\frac{1}{p} + \frac{1}{q} = 1$. For any (say) continuous function f on Δ, let us denote by $\Delta(f, \varepsilon)$ the set $\{ \mathbf{x} \in \Delta : |f(\mathbf{x})| \geq \|f\|_\infty - \varepsilon \}$, and take for g the characteristic function of $\Delta(f, \varepsilon)$. We obtain

$$(\|f\|_\infty - \varepsilon)(\operatorname{vol} \Delta(f, \varepsilon))^{1/p} \leq \|f\|_p . \tag{1}$$

Let $\mathbf{a}$ be a point of Δ where the maximum of Q_G is attained. Since $\frac{\partial Q_G}{\partial x_i} = \sum_{(i,j) \in E(G)} x_j$ we know that $0 \leq \frac{\partial Q_G}{\partial x_i} \leq 1$ for $\mathbf{x} \in \Delta$. Since Δ is convex, we conclude that for any $\mathbf{x} \in \Delta$,

$$0 \leq Q_G(\mathbf{a}) - Q_G(\mathbf{x}) \leq \sum_{i=1}^{n} |a_i - x_i| .$$

Thus $\Delta(Q_G, \varepsilon)$ contains the set $C_\varepsilon = \{ \mathbf{x} \in \Delta : \sum_{i=1}^n |a_i - x_i| < \varepsilon \}$. We claim that $\operatorname{vol}(C_\varepsilon) \geq (\frac{\varepsilon}{4})^{n-1}$. This claim proves the left inequality of the lemma when we apply it to (1).

Consider the dilated simplex $\frac{\varepsilon/2}{1+\varepsilon/2}\Delta$ and the translated set $P_\varepsilon = \frac{\mathbf{a}}{1+\varepsilon/2} + \frac{\varepsilon/2}{1+\varepsilon/2}\Delta$. Clearly P_ε is contained in Δ. Moreover, for $\mathbf{x} \in P_\varepsilon$, we have $\sum_{i=1}^n |a_i - x_i| \leq \frac{\varepsilon}{1+\varepsilon/2} \leq \varepsilon$, hence P_ε is contained in C_ε. Since $\operatorname{vol}(\Delta) = 1$ for the normalized measure, the volume of P_ε is equal to $(\frac{\varepsilon/2}{1+\varepsilon/2})^{n-1}$. Hence $\operatorname{vol}(P_\varepsilon) \geq (\varepsilon/4)^{n-1}$. This finishes the proof. □

Proof of Lemma 6. In the inequalities of Lemma 7, we substitute the relation $\|Q_G\|_\infty = \frac{1}{2}(1 - \frac{1}{\omega(G)})$, given by Motzkin–Straus's theorem (Theorem 5). We obtain

$$(\frac{1}{2}(1 - \frac{1}{\omega(G)}) - \varepsilon)(\varepsilon/4)^{\frac{n-1}{p}} \leq \|Q_G\|_p \leq \frac{1}{2}(1 - \frac{1}{\omega(G)}).$$

Let us rewrite these inequalities as

$$\frac{1}{1 - 2\|Q_G\|_p} \leq \omega(G) \leq \frac{1}{1 - \frac{2\|Q_G\|_p}{(\varepsilon/4)^{(n-1)/p}} - 2\varepsilon}. \tag{2}$$

We only need to prove that for $\varepsilon = \frac{1}{8n^2}$ and $p \geq 4(e-1)n^3 \ln(32n^2)$ we have

$$0 \leq L(p) := \frac{1}{1 - \frac{2\|Q_G\|_p}{(\varepsilon/4)^{\frac{n-1}{p}}} - 2\varepsilon} - \frac{1}{1 - 2\|Q_G\|_p} < 1. \tag{3}$$

Let us write

$$\delta_p = \|Q_G\|_p \left(\frac{1}{(\varepsilon/4)^{\frac{n-1}{p}}} - 1\right) = \|Q_G\|_p \left((32n^2)^{\frac{n-1}{p}} - 1\right).$$

Thus $L(p)$ in Equation 3 becomes now

$$L(p) = \frac{1}{1 - 2\|Q_G\|_p} \left(\frac{1}{1 - 2\frac{\delta_p + \varepsilon}{1 - 2\|Q_G\|_p}} - 1\right). \tag{4}$$

Since $\|Q_G\|_p \leq \frac{1}{2}(1 - \frac{1}{\omega(G)}) \leq \frac{1}{2}$, we have a bound for δ_p

$$0 \leq \delta_p \leq \frac{1}{2}((32n^2)^{\frac{n-1}{p}} - 1).$$

Let $A = (\frac{4}{\varepsilon})^{n-1} = (32n^2)^{n-1}$. Since we assumed $p \geq 4(e-1)n^3 \ln(32n^2)$, we have $0 \leq \frac{\ln A}{p} < 1$, hence $0 \leq A^{1/p} - 1 < (e-1)\frac{\ln A}{p}$. We obtain

$$0 \leq \delta_p \leq \frac{e-1}{2} \frac{(n-1)\log(32n^2)}{p} \leq \frac{1}{8n^2}.$$

Since $\omega(G) \leq n$, we have $1 - 2\|Q_G\|_p \geq 1/n$. Hence we have

$$\frac{2(\delta_p + \varepsilon)}{1 - 2\|Q_G\|_p} \leq \frac{1}{2n} \leq \frac{1}{2}.$$

Finally for any number $0 < \alpha < 1/2$ we have $\frac{1}{1-\alpha} < 1 + 2\alpha$, hence applying this fact to Equation 4 with $\alpha = 2\frac{\delta_p + \varepsilon}{1 - 2\|Q_G\|_p}$ we get

$$L(p) < \frac{1}{1 - 2\|Q_G\|_p} \left(\frac{4(\delta + \varepsilon)}{1 - 2\|Q_G\|_p}\right) \leq 4n^2(\delta_p + \varepsilon) \leq 1.$$

This proves Equation 3 and the lemma. □

Proof of Theorem 1. The problem of deciding whether the clique number $\omega(G)$ of a graph G is greater than a given number K is a well-known NP-complete problem [18]. From Lemma 6 we see that checking this is the same as checking that for $p = 4(e-1)n^3 \ln(32n^2)$ the integral part of $\int_\Delta (Q_G)^p \, \mathrm{d}m$ is less than K^p. Note that the polynomial $Q_G(\mathbf{x})^p$ is a power of a quadratic form and can be encoded as a SIU straight-line program of length $\mathrm{O}(n^3 \log n \cdot |E(G)|)$. If the computation of the integral $\int_\Delta f \, \mathrm{d}m$ of a polynomial f could be done in polynomial time in the input size of f, we could then verify the desired inequality in polynomial time as well. □

3.2. **An extension of a formula of Brion.** In this section, we obtain several expressions for the integrals $\int_\Delta e^\ell \, \mathrm{d}m$ and $\int_\Delta \ell_1^{M_1} \cdots \ell_D^{M_D} \, \mathrm{d}m$, where $\Delta \subset \mathbb{R}^n$ is a simplex and ℓ, ℓ_j are linear forms on $\mathbb{R}^n$. The first formula, (5) in Lemma 8, is obtained by elementary iterated integration on the standard simplex. It leads to a computation of the integral $\int_\Delta \ell_1^{M_1} \cdots \ell_D^{M_D} \, \mathrm{d}m$ in terms of the Taylor expansion of a certain

analytic function associated to Δ (Corollary 11), hence to a proof of the complexity result of Theorem 2.

In the case of one linear form ℓ which is regular, we recover in this way the "short formula" of Brion as Corollary 12. This result was first obtained by Brion as a particular case of his theorem on polyhedra [13].

Lemma 8. *Let Δ be the simplex that is the convex hull of $(d+1)$ affinely independent vertices $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{d+1}$ in $\mathbb{R}^n$, and let ℓ be an arbitrary linear form on $\mathbb{R}^n$. Then*

$$\int_\Delta e^\ell \, \mathrm{d}m = d! \operatorname{vol}(\Delta, \mathrm{d}m) \sum_{\mathbf{k}\in\mathbb{N}^{d+1}} \frac{\langle \ell, \mathbf{s}_1\rangle^{k_1} \cdots \langle \ell, \mathbf{s}_{d+1}\rangle^{k_{d+1}}}{(|\mathbf{k}|+d)!}, \tag{5}$$

where $|\mathbf{k}| = \sum_{j=1}^{d+1} k_j$.

Proof. Using an affine change of variables, it is enough to prove (5) when Δ is the d-dimensional standard simplex $\Delta_{st} \subset \mathbb{R}^d$ defined by

$$\Delta_{st} = \left\{ \mathbf{x} \in \mathbb{R}^d : x_i \geq 0, \sum_{i=1}^{d} x_i \leq 1 \right\}.$$

The volume of Δ_{st} is equal to $\frac{1}{d!}$. In the case of Δ_{st}, the vertex $\mathbf{s}_j$ is the basis vector $\mathbf{e}_j$ for $1 \leq j \leq d$ and $s_{d+1} = 0$. Let $\langle \ell, x\rangle = \sum_{j=1}^{d} a_j x_j$. Then (5) becomes

$$\int_{\Delta_{st}} e^{a_1x_1+\cdots+a_dx_d} \, \mathrm{d}\mathbf{x} = \sum_{\mathbf{k}\in\mathbb{N}^d} \frac{a_1^{k_1}\cdots a_d^{k_d}}{(|\mathbf{k}|+d)!}.$$

We prove it by induction on d. For $d = 1$, we have

$$\int_0^1 e^{ax} \, \mathrm{d}x = \frac{e^a - 1}{a} = \sum_{k\geq 0} \frac{a^k}{(k+1)!}.$$

Let $d > 1$. We write

$$\int_{\Delta_{st}} e^{a_1x_1+\cdots+a_dx_d} \, \mathrm{d}\mathbf{x} = \int_0^1 e^{a_dx_d} \left(\int_{\substack{x_j\geq 0 \\ x_1+\cdots+x_{d-1}\leq 1-x_d}} e^{a_1x_1+\cdots+a_{d-1}x_{d-1}} \, \mathrm{d}x_1 \ldots \mathrm{d}x_{d-1} \right) \mathrm{d}x_d.$$

By the induction hypothesis and an obvious change of variables, the inner integral is equal to

$$(1-x_d)^{d-1} \sum_{\mathbf{k}\in\mathbb{N}^{d-1}} (1-x_d)^{|\mathbf{k}|} \frac{a_1^{k_1}\cdots a_{d-1}^{k_{d-1}}}{(|\mathbf{k}|+d-1)!}.$$

The result now follows from the relation

$$\int_0^1 \frac{(1-x)^p}{p!} e^{ax} \, \mathrm{d}x = \sum_{k\geq 0} \frac{a^k}{(k+p+1)!}.$$

□

Remark 9. Let us replace ℓ by $t\ell$ in (5) and expand in powers of t. We obtain the following formula.

$$\int_\Delta \ell^M \,\mathrm{d}m = d!\,\mathrm{vol}(\Delta,\,\mathrm{d}m)\frac{M!}{(M+d)!}\sum_{\mathbf{k}\in\mathbb{N}^{d+1},|\mathbf{k}|=M}\langle\ell,\mathbf{s}_1\rangle^{k_1}\cdots\langle\ell,\mathbf{s}_{d+1}\rangle^{k_{d+1}}. \tag{6}$$

This relation is a particular case of a result of Lasserre and Avrachenkov, Proposition 18, as we will explain in section 4.3 below.

Theorem 10. *Let Δ be the simplex that is the convex hull of $(d+1)$ affinely independent vertices $\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_{d+1}$ in $\mathbb{R}^n$.*

$$\sum_{M\in\mathbb{N}} t^M\frac{(M+d)!}{M!}\int_\Delta \ell^M \,\mathrm{d}m = d!\,\mathrm{vol}(\Delta,\,\mathrm{d}m)\frac{1}{\prod_{j=1}^{d+1}(1-t\langle\ell,\mathbf{s}_j\rangle)}. \tag{7}$$

Proof. We apply Formula (6). Summing up from $M=0$ to ∞, we recognize the expansion of the right-hand side of (7) into a product of geometric series:

$$\sum_{M\in\mathbb{N}} t^M\frac{(M+d)!}{M!}\int_\Delta \ell^M \,\mathrm{d}m = d!\,\mathrm{vol}(\Delta,\,\mathrm{d}m)\sum_{M\in\mathbb{N}} t^M\sum_{\mathbf{k}\in\mathbb{N}^{d+1}|,\mathbf{k}|=M}\langle\ell,\mathbf{s}_1\rangle^{k_1}\cdots\langle\ell,\mathbf{s}_{d+1}\rangle^{k_{d+1}}.$$

□

Theorem 10 has an extension to the integration of a product of powers of *several* linear forms. The following formula is implemented in our Maple program `duality.mpl`, see Table 8

Corollary 11. *Let $\ell_1,\ldots,\ell_D$ be D linear forms on $\mathbb{R}^n$. We have the following Taylor expansion:*

$$\sum_{\mathbf{M}\in\mathbb{N}^D} t_1^{M_1}\cdots t_D^{M_D}\,\frac{(|\mathbf{M}|+d)!}{d!\,\mathrm{vol}(\Delta,\,\mathrm{d}m)}\int_\Delta\frac{\ell_1^{M_1}\cdots\ell_D^{M_D}}{M_1!\ldots M_D!}\,\mathrm{d}m = \frac{1}{\prod_{i=1}^{d+1}(1-t_1\langle\ell_1,\mathbf{s}_i\rangle-\cdots-t_D\langle\ell_D,\mathbf{s}_i\rangle)}. \tag{8}$$

Proof. Replace $t\ell$ with $t_1\ell_1+\cdots+t_D\ell_D$ in (7) and take the expansion in powers $t_1^{M_1}\cdots t_D^{M_D}$. □

From Theorem 10, we obtain easily the "short formula" of Brion, in the case of a simplex.

Corollary 12 (Brion)**.** *Let Δ be as in the previous theorem. Let ℓ be a linear form which is regular w.r.t. Δ, i.e., $\langle\ell,\mathbf{s}_i\rangle\neq\langle\ell,\mathbf{s}_j\rangle$ for any pair*

$i \neq j$. Then we have the following relations.

$$\int_\Delta \ell^M \,\mathrm{d}m = d!\,\mathrm{vol}(\Delta,\,\mathrm{d}m)\frac{M!}{(M+d)!}\Big(\sum_{i=1}^{d+1}\frac{\langle \ell, \mathbf{s}_i\rangle^{M+d}}{\prod_{j\neq i}\langle \ell, \mathbf{s}_i - \mathbf{s}_j\rangle}\Big). \tag{9}$$

$$\int_\Delta e^\ell \,\mathrm{d}m = d!\,\mathrm{vol}(\Delta,\,\mathrm{d}m)\sum_{i=1}^{d+1}\frac{e^{\langle \ell, \mathbf{s}_i\rangle}}{\prod_{j\neq i}\langle \ell, \mathbf{s}_i - \mathbf{s}_j\rangle}. \tag{10}$$

Proof. We consider the right-hand side of (7) as a rational function of t. The poles $t = 1/\langle \ell, \mathbf{s}_i\rangle$ are simple precisely when ℓ is regular. In this case, we obtain (9) by taking the expansion into partial fractions. The second relation follows immediately by expanding e^ℓ. □

When ℓ is regular, Brion's formula is very short, it is a sum of $d+1$ terms. When ℓ is not regular, the expansion of (7) into partial fractions leads to an expression of the integral as a sum of residues. Let $K \subseteq \{1, \ldots, d+1\}$ be an index set of the different poles $t = 1/\langle \ell, s_k\rangle$, and for $k \in K$ let m_k denote the order of the pole, i.e.,

$$m_k = \#\big\{\, i \in \{1, \ldots, d+1\} : \langle \ell, s_i\rangle = \langle \ell, s_k\rangle \,\big\}.$$

With this notation, we have the following formula, which is implemented in our Maple program `waring.mpl`, see Tables 5 and 6.

Corollary 13.

$$\int_\Delta \ell^M \,\mathrm{d}m = d!\,\mathrm{vol}(\Delta,\,\mathrm{d}m)\frac{M!}{(M+d)!}\sum_{k\in K}\mathrm{Res}_{\varepsilon=0}\frac{(\varepsilon + \langle \ell, \mathbf{s}_k\rangle)^{M+d}}{\varepsilon^{m_k}\prod_{\substack{i\in K\\ i\neq k}}(\varepsilon + \langle \ell, \mathbf{s}_k - \mathbf{s}_i\rangle)^{m_i}} \tag{11}$$

Remark 14. It is worth remarking that Corollaries 12 and 13 can be seen as a particular case of the localization theorem in equivariant cohomology (see for instance [8]), although we did not use this fact and instead gave a simple direct calculation. In our situation, the variety is the complex projective space $\mathbb{CP}^d$, with action of a d-dimensional torus, such that the image of the moment map is the simplex Δ. Brion's formula corresponds to the generic case of a one-parameter subgroup acting with isolated fixed points. In the degenerate case when the set of fixed points has components of positive dimension, the polar parts in (11) coincide with the contributions of the components to the localization formula.

A formula equivalent to Corollary 13 appears already in [3] (3.2).

3.3. **Polynomial time algorithm for polynomial functions of a fixed number of linear forms.**

Proof of Theorem 2. We now present an algorithm which, given a polynomial of the particular form $f(\langle \ell_1, \mathbf{x}\rangle, \cdots, \langle \ell_D, \mathbf{x}\rangle)$ where f is a polynomial depending on a fixed number D of variables, and $\langle \ell_j, \mathbf{x}\rangle = L_{j1}x_1 + \cdots + L_{jn}x_n$, for $j = 1, \ldots, D$, are linear forms on $\mathbb{R}^n$, computes its integral on a simplex, in time polynomial on the input data. This algorithm relies on Corollary 11.

The number of monomials of degree M in D variables is equal to $\binom{M+D-1}{D-1}$. Therefore, when D is fixed, the number of monomials of degree at most M in D variables is $\mathrm{O}(M^D)$. When the number of variables D of a straight-line program Φ is fixed, it is possible to compute a sparse or dense representation of the polynomial represented by Φ in polynomial time, by a straight-forward execution of the program. Indeed, all intermediate results can be stored as sparse or dense polynomials with $\mathrm{O}(M^D)$ monomials. Since the program Φ is single-intermediate-use, the binary encoding size of all coefficients of the monomials can be bounded polynomially by the input encoding size. Thus it is enough to compute the integral of a monomial,

$$\int_\Delta \langle \ell_1, \mathbf{x}\rangle^{M_1} \cdots \langle \ell_D, \mathbf{x}\rangle^{M_D} \, \mathrm{d}m. \tag{12}$$

From Corollary 11, it follows that

$$\frac{(|\mathbf{M}| + d)!}{d! \operatorname{vol}(\Delta, \mathrm{d}m)} \int_\Delta \frac{\ell_1^{M_1} \cdots \ell_D^{M_D}}{M_1! \cdots M_D!} \, \mathrm{d}m$$

is the coefficient of $t_1^{M_1} \cdots t_D^{M_D}$ in the Taylor expansion of

$$\frac{1}{\prod_{i=1}^{d+1}(1 - t_1\langle \ell_1, \mathbf{s}_i\rangle - \cdots - t_D\langle \ell_D, \mathbf{s}_i\rangle)}.$$

Since D is fixed, this coefficient can be computed in time polynomial with respect to $\mathbf{M}$ and the input data, by multiplying series truncated at degree $|\mathbf{M}|$, as explained in Lemma 4.

Finally, $\operatorname{vol}(\Delta, \mathrm{d}m)$ needs to be computed. If $\Delta = \operatorname{conv}\{\mathbf{s}_1, \ldots, \mathbf{s}_{d+1}\}$ is full-dimensional ($d = n$), we can do so by computing the determinant of the matrix formed by difference vectors of the vertices:

$$\operatorname{vol}(\Delta, \mathrm{d}m) = \frac{1}{n!} \left|\det(\mathbf{s}_1 - \mathbf{s}_{n+1}, \ldots, \mathbf{s}_n - \mathbf{s}_{n+1})\right|.$$

If Δ is lower-dimensional, we first compute a basis $B \in \mathbb{Z}^{n\times d}$ of the intersection lattice $\Lambda = \operatorname{lin}(\Delta) \cap \mathbb{Z}^n$. This can be done in polynomial time by applying an efficient algorithm for computing the Hermite normal form [19]. Then we express each difference vector $\mathbf{v}_i = \mathbf{s}_i - \mathbf{s}_{d+1} \in \operatorname{lin}(\Delta)$ for $i = 1, \ldots, d$ using the basis B as $\mathbf{v}_i = B\mathbf{v}'_i$, where $\mathbf{v}_i \in \mathbb{Q}^d$. We obtain

$$\operatorname{vol}(\Delta, \mathrm{d}m) = \frac{1}{d!} \left|\det(\mathbf{v}'_1, \ldots, \mathbf{v}'_d)\right|,$$

thus the volume computation is reduced to the calculation of a determinant. This finishes the proof of Theorem 2. □

3.4. **Polynomial time algorithms for polynomials of fixed degree.** In the present section, we assume that the total degree of the input polynomial f we wish to integrate is a constant M.

Proof of Corollary 3. First of all, when the formal degree M of a straight-line program Φ is fixed, it is possible to compute a sparse or dense representation of the polynomial represented by Φ in polynomial time, by a straight-forward execution of the program. Indeed, all intermediate results can be stored as sparse or dense polynomials with $O(n^M)$ monomials. Since the program Φ is single-intermediate-use, the binary encoding size of all coefficients of the monomials can be bounded polynomially by the input encoding size.

Now, the key observation is that a monomial of degree at most M depends effectively on $D \leq M$ variables $x_{i_1}, \ldots, x_{i_D}$, thus it is of the form

$$\ell_1^{M_1} \cdots \ell_D^{M_D},$$

where the linear forms $\ell_j(\mathbf{x}) = x_{i_j}$ are the coordinates that effectively appear in the monomial. Thus, Corollary 3 follows immediately from Theorem 2. This method is implemented in our Maple program `duality.mpl`, see Tables 8 and 12. □

Remark 15. The relations in Corollary 12 can be interpreted as equalities between meromorphic functions of ℓ. The right-hand side is a sum of meromorphic functions whose poles cancel out, so that the sum is actually analytic. We derive from this another polynomial time algorithm for computating the integral

$$\int_\Delta x_1^{m_1} \cdots x_d^{m_d} \, \mathrm{d}m.$$

More precisely, let us write $\langle \ell, \mathbf{x} \rangle = y_1 x_1 + \cdots + y_d x_d$. Then the integral $\int_\Delta x_1^{m_1} \cdots x_d^{m_d} \, \mathrm{d}m$ is the coefficient of $\frac{y_1^{m_1} \ldots y_d^{m_d}}{m_1! \cdots m_d!}$ in the Taylor expansion of $\int_\Delta e^{y_1 x_1 + \ldots y_d x_d} \, \mathrm{d}m$. We compute it by taking the expansion of each of the terms of the right hand-side of Equation (10) into an iterated Laurent series with respect to the variables $x_1, \ldots, x_d$. This method is implemented in our Maple program `iterated-laurent.mpl`, see Tables 11 and 14.

In the following, we give another proof of Corollary 3, based on decompositions of polynomials as sums of powers of linear forms.

Alternative proof of Corollary 3. From Corollaries 12 and 13, we derive another efficient algorithm, as follows. The key idea now is that one

can decompose the polynomial f as a sum $f := \sum_\ell c_\ell \ell_j^M$ with at most 2^M terms in the sum. We use the well-known identity

$$x_1^{M_1} x_2^{M_2} \cdots x_n^{M_n} = \frac{1}{|\mathbf{M}|!} \sum_{0 \le p_i \le M_i} (-1)^{|\mathbf{M}|-(p_1+\cdots+p_n)} \binom{M_1}{p_1} \cdots \binom{M_n}{p_n} (p_1 x_1 + \cdots + p_n x_n)^{|\mathbf{M}|}, \tag{13}$$

where $|\mathbf{M}| = M_1 + \cdots + M_n \le M$.

In the implementation of this method, we may group together proportional linear forms. The number $F(n, M)$ of primitive vectors $(p_1, \ldots, p_n)$ which appear in the decomposition of a polynomial of total degree $\le M$ is given by the following closed formula. [1].

Lemma 16. *Let*

$$F(n, M) = \mathrm{Card}(\{(p_1, \ldots, p_n) \in \mathbb{N}^n, \gcd(p_1, \ldots, p_n) = 1, 1 \le \sum_i p_i \le M\}).$$

Then

$$F(n, M) = \sum_{d=1}^{M} \mu(d) \left(\binom{n + [\frac{M}{d}]}{n} - 1 \right), \tag{14}$$

where $\mu(d)$ is the Möbius function.

When M is fixed and $n \to \infty$ we have

$$F(n, M) = \frac{n^M}{M!} + \mathrm{O}(n^{M-1}).$$

Proof. Let $G(n, M) = \mathrm{Card}(\{(p_1, \ldots, p_n) \in \mathbb{N}^n, 1 \le \sum p_i \le M\})$. By grouping together the vectors $(p_1, \ldots, p_n)$ with a given gcd d, we obtain

$$G(n, M) = \sum_{d=1}^{M} F(n, [\frac{M}{d}]).$$

Moreover the number of all integral vectors $(p_1, \ldots, p_n) \in \mathbb{N}^n$ such that $\sum_i p_i \le M$ is equal to the binomial coefficient $\binom{n+M}{n}$. When we omit the zero-vector we obtain

$$G(n, M) = \binom{n + M}{n} - 1.$$

Then we obtain (14) by applying the second form of Möbius inversion formula. The asymptotics follow easily from (14). □

Thus Formula (13), together with Corollary 13, give another polynomial time algorithm for integrating a polynomial of fixed degree. It is implemented in our Maple program `waring.mpl`, see Tables 6 and 10. □

[1] Lemma 16 was kindly supplied by Christophe Margerin.

The problem of finding a decomposition with the smallest possible number of summands is known as the *polynomial Waring problem*. Alexander and Hirschowitz solved the generic problem (see [1], and [11] for an extensive survey).

Theorem 17. *The smallest integer $r(M,n)$ such that a generic homogeneous polynomial of degree M in n variables is expressible as the sum of $r(M,n)$ M-th powers of linear forms is given by*

$$r(M,n) = \left\lceil \frac{\binom{n+M-1}{M}}{n} \right\rceil,$$

with the exception of the cases $r(3,5) = 8$, $r(4,3) = 6$, $r(4,4) = 10$, $r(4,5) = 15$, and $M = 2$, where $r(2,n) = n$.

An algorithm for decomposing a given polynomial into the smallest possible number of powers of linear forms can be found in [10].

In the extreme case, when the polynomial f happens to be the power of one linear form ℓ, one should certainly avoid applying the above decomposition formula to each of the monomials of f. We remark that, when the degree is fixed, we can decide in polynomial time whether a polynomial f, given in sparse or dense monomial representation, is a power of a linear form ℓ, and, if so, construct such a linear form.

4. Other algorithms for integration and extensions to other polytopes

We conclude with a discussion of how to extend integration to other polytopes and a review of the complexity of other methods to integrate polynomials over polytopes.

4.1. A formula of Lasserre–Avrachenkov.

Another nice formula is the Lasserre–Avrachenkov formula for the integration of a homogeneous polynomial [22] on a simplex. As we explain below, this yields a polynomial-time algorithm for the problem of integrating a polynomial of fixed degree over a polytope in varying dimension, thus providing an alternative proof of Corollary 3.

Proposition 18 ([22])**.** *Let H be a symmetric multilinear form defined on $(\mathbb{R}^d)^M$. Let $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{d+1}$ be the vertices of a d-dimensional simplex Δ. Then one has*

$$\int_\Delta H(\mathbf{x}, \mathbf{x}, \ldots, \mathbf{x}) d\mathbf{x} = \frac{\operatorname{vol}(\Delta)}{\binom{M+d}{M}} \sum_{1 \le i_1 \le i_2 \le \cdots i_M \le d+1} H(\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \ldots, \mathbf{s}_{i_M}). \quad (15)$$

Remark 19. By reindexing the summation in (15), as H is symmetric, we obtain

$$\int_\Delta H(\mathbf{x},\mathbf{x},\dots,\mathbf{x})d\mathbf{x} = \frac{\mathrm{vol}(\Delta)}{\binom{M+d}{M}} \sum_{k_1+\cdots+k_{d+1}=M} H(\mathbf{s}_1,\dots,\mathbf{s}_1,\dots,\mathbf{s}_{d+1},\dots,\mathbf{s}_{d+1}), \quad (16)$$

where $\mathbf{s}_1$ is repeated k_1 times, $\mathbf{s}_2$ is repeated k_2 times, etc. When H is of the form $H = \prod_{i=1}^M \langle \ell, \mathbf{x}_i\rangle$, for a single linear form ℓ, then (16) coincides with Formula (6) in Remark 9.

Now any polynomial f which is homogeneous of degree M can be written as $f(\mathbf{x}) = H_f(\mathbf{x},\mathbf{x},\dots,\mathbf{x})$ for a unique multilinear form H_f. If $f = \ell^M$ then $H_f = \prod_{i=1}^M \langle \ell, \mathbf{x}_i\rangle$. Thus for fixed M the computation of H_f can be done by decomposing f into a linear combination of powers of linear forms, as we did in the proof of Corollary 3. Alternatively one can use the well-known polarization formula,

$$H_f(\mathbf{x}_1,\dots,\mathbf{x}_M) = \frac{1}{2^M M!} \sum_{\boldsymbol{\varepsilon}\in\{\pm 1\}^M} \varepsilon_1\varepsilon_2\cdots\varepsilon_M f\Big(\sum_{i=1}^M \varepsilon_i \mathbf{x}_i\Big), \quad (17)$$

Thus from (15) we get the following corollary.

Corollary 20. *Let f be a homogeneous polynomial of degree M in d variables, and let $\mathbf{s}_1,\mathbf{s}_2,\dots,\mathbf{s}_{d+1}$ be the vertices of a d-dimensional simplex Δ. Then*

$$\int_\Delta f(\mathbf{y})\,\mathrm{d}\mathbf{y} = \frac{\mathrm{vol}(\Delta)}{2^M M!\binom{M+d}{M}} \sum_{1\le i_1\le i_2\le\cdots\le i_M\le d+1} \sum_{\boldsymbol{\varepsilon}\in\{\pm 1\}^M} \varepsilon_1\varepsilon_2\cdots\varepsilon_M f\Big(\sum_{k=1}^M \varepsilon_k \mathbf{s}_{i_k}\Big). \quad (18)$$

We remark that when we fix the degree M of the homogeneous polynomial f, the length of the polarization formula (thus the length of the second sum in (18) is a constant. The length of the first sum in (18) is $\mathrm{O}(n^M)$. Thus, for fixed degree in varying dimension, we obtain another polynomial-time algorithm for integrating over a simplex.

4.2. **Traditional conversion of the integral as iterated univariate integrals.** Let $P \subseteq \mathbb{R}^d$ be a full-dimensional polytope and f a polynomial. The traditional method we teach our calculus students to compute multivariate integrals over a bounded region requires them to write the integral $\int_P f\,\mathrm{d}m$ is a sum of sequences of one-dimensional integrals

$$\sum_{j=1}^K \int_{a_{1j}}^{b_{1j}} \int_{a_{2j}}^{b_{2j}} \cdots \int_{a_{dj}}^{b_{dj}} f\,\mathrm{d}x_{i_1}\,\mathrm{d}x_{i_2}\dots\mathrm{d}x_{i_d} \quad (19)$$

for which we know the limits of integration a_{ij}, b_{ij} explicitly. The problem of finding the limits of integration and the sum has interesting complexity related to the well-known Fourier-Motzkin elimination method (see Chapter One in [30] for a short introduction).

Given a system of linear inequalities $A\mathbf{x} \leq \mathbf{b}$, describing a polytope $P \subset \mathbb{R}^d$, Fourier–Motzkin elimination is an algorithm that computes a new larger system of inequalities $\hat{A}\mathbf{x} \leq \hat{\mathbf{b}}$ with the property that those inequalities that do not contain the variable x_d describe the projection of P into the hyperplane $x_d = 0$. We will not explain the details, but Fourier-Motzkin elimination is quite similar to Gaussian elimination in the sense that the main operations necessary to eliminate the last variable x_d require to rearrange, scale, and add rows of the matrix $(A, \mathbf{b})$, but unlike Gaussian elimination, new inequalities are added to the system.

It was first observed by Schechter [29] that Fourier-Motzkin elimination provides a way to generate the traditional iterated integrals. More precisely, let us call P_d the projection of P into the the hyperplane $x_d = 0$. Clearly when integrating over a polytopal region we expect that the limits of integration will be affine functions. From the output of Fourier-Motzkin $\hat{A}\mathbf{x} \leq \hat{\mathbf{b}}$, we have $\mathbf{x} \in P$ if and only if $(x_1, \ldots, x_{d-1}) \in P_d$ and for the first $k + r$ inequalites of the system

$$x_d \leq \hat{b}_i - \sum_{j=1}^{d-1} \hat{a}_{ij} x_j = A_i^{\mathrm{u}}(x_1, \ldots, x_{d-1})$$

for $i = 1, \ldots k$ as well as

$$x_d \geq \hat{b}_{k+i} - \sum_{j=1}^{d-1} \hat{a}_{k+ij} x_j, = A_i^{\mathrm{l}}(x_1, \ldots, x_{d-1})$$

for $i = 1, \ldots, r$. Then, if we define

$$m(x_1, \ldots, x_d) = \max\{A_j^{\mathrm{l}}(x_1, \ldots, x_{d-1}),\ j = 1, \ldots, r\}$$

and

$$M(x_1, \ldots, x_d) = \min\{A_j^{\mathrm{u}}(x_1, \ldots, x_{d-1}),\ j = 1, \ldots, r\},$$

we can write

$$\int_P f(\mathbf{x})\, \mathrm{d}m = \int_{P_d} \int_m^M f(\mathbf{x})\, \mathrm{d}x_1\, \mathrm{d}x_2 \cdots \mathrm{d}x_d$$

Finally the convex polytope P_d can be decomposed into polyhedral regions where the functions m, M become simply affine functions from among the list. Since the integral is additive we get an expression

$$\int_P f(\mathbf{x})\, \mathrm{d}m = \sum_{i,j} \int_{P_d^{ij}} \int_{A_j^{\mathrm{l}}}^{A_j^{\mathrm{u}}} f(\mathbf{x})\, \mathrm{d}x_1\, \mathrm{d}x_2 \cdots \mathrm{d}x_d.$$

Finally by repeating the elimination of variables we recover the full iterated list in (19). As it was observed in [29], this algorithm is unfortunately not efficient because the iterated Fourier-Motzkin elimination procedure can produce exponentially many inequalities for the description of the projection (when the dimension d varies). Thus the number of summands considered can in fact grow exponentially.

4.3. **Two formulas for integral computation.** We would like to review two formulas that are nice and could speed up computation in particular cases although they do not seem to yield efficient algorithms just on their own.

First, one may reduce the computation of $\int_P f \, \mathrm{d}m$ to integrals on the facets of P, by applying Stokes formula. We must be careful to use a rational Lebesgue measure on each facet. As shown in ([4]), we have the following result.

Theorem 21. *Let $\{F_i\}_{i=1,\dots,m}$ be the set of facets of a full-dimensional polytope $P \subseteq \mathbb{R}^n$. For each i, let $\mathbf{n}_i$ be a rational vector which is transverse to the facet F_i and pointing outwards P and let $\mathrm{d}\mu_i$ be the Lebesgue measure on the affine hull of F_i which is defined by contracting the standard volume form of $\mathbb{R}^n$ with $\mathbf{n}_i$. Then*

$$I_P(\mathbf{a}) = \int_P e^{\langle \mathbf{a},\mathbf{x}\rangle} \, \mathrm{d}\mathbf{x} = \frac{1}{\langle \mathbf{a},\mathbf{y}\rangle} \sum_{i=1}^{m} \langle \mathbf{n}_i,\mathbf{y}\rangle \int_{F_i} e^{\langle \mathbf{a},\mathbf{x}\rangle} \, \mathrm{d}\mu_i$$

for all $\mathbf{a} \in \mathbb{C}^n$ and $\mathbf{y} \in \mathbb{R}^n$ such that $\langle \mathbf{y},\mathbf{a}\rangle \neq 0$.

It is clear that, by considering the expansion of the analytic function $\int_P e^{\langle \mathbf{a},\mathbf{x}\rangle} \, \mathrm{d}\mathbf{x}$, we can again obtain an analogous result for polynomials. An alternative proof was provided by [21]. The above theorem, however, does not necessarily reduce the computational burden because, depending on the representation of the polytope, the number of facets can be large and also the facets themselves can be complicated polytopes. Yet, together with our results we obtain the following corollary for two special cases.

Corollary 22. *There is a polynomial-time algorithm for the following problem. Input:*

- (I_1) *the dimension $n \in \mathbb{N}$ in unary encoding,*
- (I_2) *a list of rational vectors in binary encoding, namely*
  - (i) *either vectors $(\mathbf{h}_1, h_{1,0}), \dots, (\mathbf{h}_m, h_{m,0}) \in \mathbb{Q}^{n+1}$ that describe the facet-defining inequalities $\langle \mathbf{h}_i,\mathbf{x}\rangle \leq h_{i,0}$ of a* simplicial *full-dimensional rational polytope P,*
  - (ii) *or vectors $\mathbf{s}_1, \dots, \mathbf{s}_N \in \mathbb{Q}^n$ that are the vertices of a* simple *full-dimensional rational polytope P,*
- (I_3) *a rational vector $\mathbf{a} \in \mathbb{Q}^n$ in binary encoding,*
- (I_4) *an exponent $M \in \mathbb{N}$ in unary encoding.*

Output, in binary encoding,

(O_1) *the rational number*

$$\int_P f(\mathbf{x})\,\mathrm{d}m \quad \textit{where} \quad f(\mathbf{x}) = \langle \mathbf{a}, \mathbf{x}\rangle^M$$

and where $\mathrm{d}m$ *is the standard Lebesgue measure on* $\mathbb{R}^n$.

Proof. In the case (i) of simplicial polytopes P given by facet-defining inequalities, we can use linear programming to compute in polynomial time a V-representation for each simplex F_i that is a facet of P. By applying Theorem 21 with $t\mathbf{a}$ in place of $\mathbf{a}$ and extracting the coefficient of t^M in the Taylor expansion of the analytic function $t \mapsto I_P(t\mathbf{a})$, we obtain the formula

$$\int_P \langle \mathbf{a}, \mathbf{x}\rangle^M \,\mathrm{d}\mathbf{x} = \frac{1}{(M+1)\langle \mathbf{y}, \mathbf{a}\rangle} \sum_{i=1}^{m} \langle \mathbf{y}, \mathbf{n}_i\rangle \int_{F_i} \langle \mathbf{a}, \mathbf{x}\rangle^{M+1} \,\mathrm{d}\mu_i,$$

which holds for all $\mathbf{y} \in \mathbb{R}^n$ with $\langle \mathbf{y}, \mathbf{a}\rangle \neq 0$. It is known that a suitable $\mathbf{y} \in \mathbb{Q}^n$ can be constructed in polynomial time. The integrals on the right-hand side can now be evaluated in polynomial time using Theorem 2.

In the case (ii) of simple polytopes P given by their vertices, we make use of the fact that a variant of Brion's formula (10) actually holds for arbitrary rational polytopes. For a simple polytope P, it takes the following form.

$$\int_P \ell^M \,\mathrm{d}x = \frac{M!}{(M+n)!} \sum_{i=1}^{N} \Delta_i \frac{\langle \ell, \mathbf{s}_i\rangle^{M+n}}{\prod_{\mathbf{s}_j \in N(\mathbf{s}_i)} \langle \ell, \mathbf{s}_i - \mathbf{s}_j\rangle}, \tag{20}$$

where $N(\mathbf{s}_i)$ denotes the set of vertices adjacent to $\mathbf{s}_i$ in P, and $\Delta_i = |\det(\mathbf{s}_i - \mathbf{s}_j)_{j \in N(\mathbf{s}_i)}|$. The right-hand side is a sum of rational functions of ℓ, where the denominators cancel out so that the sum is actually polynomial. If ℓ is regular, that is to say $\langle \ell, \mathbf{s}_i - \mathbf{s}_j\rangle \neq 0$ for any i and $j \in N(\mathbf{s}_i)$, then the integral can be computed by (20) which is a very short formula. However it becomes difficult to extend the method which we used in the case of a simplex. Instead, we can do a perturbation. In (20), we replace ℓ by $\ell + \varepsilon\ell'$, where ℓ' is such that $\ell + \varepsilon\ell'$ is regular for $\varepsilon \neq 0$. The algorithm for choosing ℓ' is bounded polynomially. Then we do expansions in powers of ε as explained in Lemma 4. □

4.4. **Triangulation of arbitrary polytopes.** It is well-known that any convex polytope can be triangulated into finitely many simplices. Thus we can use our result to extend the integration of polynomials over any convex polytope. The complexity of doing it this way will directly depend on the number of simplices in a triangulation. This raises the issue of finding the smallest triangulation possible of a given polytope. Unfortunately this problem was proved to be NP-hard even for fixed dimension three (see [15]). Thus it is in general not a good idea to spend time finding the smallest triangulation possible. A cautionary

remark is that one can naively assume that triangulations help for nonconvex polyhedral regions, while in reality it does not because there exist nonconvex polyhedra that are not triangulable unless one adds new points. Deciding how many new points are necessary is an NP-hard problem [15].

5. Implementation and computational experiments

We have written Maple programs to perform some initial experiments with the three methods described in Section 3.4. The programs are available at [2].[2]

5.1. Integration of a power of a linear form and decomposition of polynomials into powers of linear forms.

5.1.1. *Decomposition of polynomials into powers of linear forms.* Table 4 shows the number $F(n, M)$ of primitive linear forms $(p_1, \dots, p_n)$ which may appear in the decomposition (13) of a polynomial of total degree $\leq M$. This number is computed using the closed formula (14).

Table 4. Decomposition of polynomials into powers of linear forms

	Degree M							
n	1	2	5	10	20	30	40	50
2	2	3	11	33	129	279	491	775
3	3	6	40	205	1381	4306	9880	18970
4	4	10	103	831	9373	41373	122349	286893
5	5	15	221	2681	49586	305836	$1.2 \cdot 10^6$	$3.3 \cdot 10^6$
8	8	36	1226	42271	$3.1 \cdot 10^6$	$4.8 \cdot 10^7$	$3.7 \cdot 10^8$	$1.9 \cdot 10^9$
10	10	55	2917	181413	$3.0 \cdot 10^7$	$8.4 \cdot 10^8$	$1.0 \cdot 10^{10}$	$7.5 \cdot 10^{10}$
15	15	120	15338	$3.3 \cdot 10^6$	$3.2 \cdot 10^9$	$3.4 \cdot 10^{11}$	$1.2 \cdot 10^{13}$	$2.1 \cdot 10^{14}$
20	20	210	52859	$3.0 \cdot 10^7$	$1.4 \cdot 10^{11}$	$4.7 \cdot 10^{13}$	$4.2 \cdot 10^{15}$	$1.6 \cdot 10^{17}$
30	30	465	324076	$8.5 \cdot 10^8$	$4.7 \cdot 10^{13}$	$1.2 \cdot 10^{17}$	$5.5 \cdot 10^{19}$	$8.9 \cdot 10^{21}$
40	40	820	$1.2 \cdot 10^6$	$1.0 \cdot 10^{10}$	$4.2 \cdot 10^{15}$	$5.5 \cdot 10^{19}$	$1.1 \cdot 10^{23}$	$6.0 \cdot 10^{25}$
50	50	1275	$3.5 \cdot 10^6$	$7.5 \cdot 10^{10}$	$1.6 \cdot 10^{17}$	$8.9 \cdot 10^{21}$	$6.0 \cdot 10^{25}$	$1.0 \cdot 10^{29}$

[2] All algorithms are implemented in the files `waring.mpl`, `iterated_laurent.mpl`, `duality.mpl`; all tables with random examples are created using procedures in `examples.mpl` and `tables.mpl`.

5.1.2. *Integration of a power of a linear form over a simplex.* We have written a Maple program which implements the method of Corollary 13 for the efficient integration of a power of one linear form over a simplex, $\int_\Delta \ell^M \, \mathrm{d}m$.[3] In a computational experiment, for a given dimension n and degree M, we picked random full-dimensional simplices Δ and random linear forms ℓ and used the Maple program to compute the integral. Table 5 shows the computation times.[4]

TABLE 5. Integration of powers of linear forms over simplices

	Degree M						
n	2	10	20	50	100	300	1000
10	0.0	0.0	0.0	0.1	0.0	0.0	0.0
20	0.1	0.2	0.1	0.2	0.1	0.2	0.2
50	1.0	1.4	1.4	1.6	1.6	1.6	1.7
100	5.1	8.4	8.7	9.2	9.5	10.0	11.0
200	36	71	84	88	97	110	120
300	150	320	400	470	520	530	
400	500						
1000							

5.1.3. *Integration of a monomial over a simplex by decomposition as sum of powers of linear forms.* Next, we tested the algorithm which computes the integral of a monomial $\mathbf{x}^\mathbf{M}$ over a simplex Δ, by decomposing it as a sum of powers of linear forms. This algorithm was discussed in Section 3.4. In our experiments, for given dimension n and total degree M, we picked 50 combinations of a random simplex Δ of dimension n and a random exponent vector $\mathbf{M} = (M_1, \ldots, M_n)$ with $\sum_{i=1}^n M_i = M$.

First we decompose a given monomial into a sum of powers of linear forms, then we integrate each summand using the Maple procedure discussed above. [5] Table 6 shows the minimum, average, and maximum computation times.

[3]The integration is done by the Maple procedure *integral_power_linear_form* in `waring.mpl`.

[4]All experiments were done with Maple 12 on Sun Fire V440 machines with UltraSPARC-IIIi processors running at 1.6 GHz. The computation times are given in CPU seconds. All experiments were subject to a time limit of 600 seconds per example.

[5]This method is implemented in the Maple procedure *integral_via_waring* in `waring.mpl`.

TABLE 6. Integration of a random monomial of prescribed degree by decomposition into a sum of powers of linear forms

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
2	0	0	0	0	0	0	0.1	0.3	0.8	6.2	0.3
	0	**0**	**0**	**0**.1	**0**.2	**0**.5	**1**.0	**1**.5	**8**.3	**38**	**90**
	0	0.1	0.1	0.1	0.4	0.8	1.9	2.6	12	59	174
3	0	0	0	0	0	0.4	0.5	0.9	4.6		
	0	**0**	**0**	**0**.2	**1**.1	**3**.0	**7**.4	**17**	**173**		
	0	0.2	0.1	0.4	2.2	7.0	19	37	440		
4	0	0	0	0.1	0.6	1.5	4.8	9.7			
	0	**0**	**0**.1	**0**.4	**3**.6	**15**	**52**	**135**			
	0	0.2	0.2	0.9	8.7	44	149	404			
5	0	0	0	0.1	0.3	4.5					
	0	**0**	**0**.1	**0**.7	**8**.8	**48**					
	0	0.2	0.3	1.9	27	195					
6	0	0	0	0.2	1.3	8.0					
	0	**0**	**0**.2	**1**.3	**24**	**144**					
	0.2	0.1	0.4	2.7	74	544					
7	0	0	0	0.5	5.9						
	0	**0**	**0**.3	**2**.1	**53**						
	0.2	0.1	0.6	5.0	152						
8	0	0	0.1	0.4	11						
	0	**0**	**0**.3	**3**.2	**72**						
	0.2	0.2	0.6	8.5	216						
10	0	0	0.2	1.5							
	0	**0**.1	**0**.4	**6**.1							
	0.3	0.2	0.8	12							
15	0	0	0.3	3.8							
	0.1	**0**.1	**1**.2	**17**							
	0.2	0.3	1.8	41							
20	0.1	0.1	0.6	4.4							
	0.1	**0**.3	**2**.2	**41**							
	0.2	0.4	2.9	73							
30	0.1	0.2	2.7	37							
	0.2	**0**.5	**5**.1	**106**							
	0.3	0.6	6.8	170							
40	0.3	0.3	5.2	93							
	0.4	**1**.1	**10**	**242**							
	0.6	1.3	12	414							
50	0.5	0.7	8.2								
	0.6	**1**.8	**17**								
	0.8	2.0	20								

5.2. **Integration of a monomial, using iterated Laurent series.** In this section, we test the implementation of the method of iterated Laurent expansion described in Remark 15 of Section 3.4.[6]

Table 7 shows the results.

TABLE 7. Integration of a random monomial of prescribed degree using iterated Laurent series.

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
2	0	0	0	0	0	0	0	0	0	0.3	0.2
	0	**0**	**0**	**0**	**0**	**0**	**0**.1	**0**.1	**0**.5	**3**.0	**8**.1
	0	0.1	0.1	0.1	0.1	0.1	0.2	0.3	1.2	7.1	30
3	0	0	0	0	0	0	0.1	0.1	0.2		
	0	**0**	**0**	**0**.1	**0**.4	**0**.5	**1**.2	**3**.4	**34**		
	0	0	0.1	0.2	6.2	3.7	7.1	11	164		
4	0	0	0	0	0.1	0.3	0.4	1.0			
	0	**0**	**0**.1	**0**.3	**1**.4	**4**.8	**16**	**39**			
	0	0.1	0.2	0.7	12	35	77	176			
5	0	0	0.1	0.1	0.1	1.4	0.2				
	0.1	**0**.1	**0**.2	**0**.7	**4**.5	**36**	**78**				
	0.1	0.1	0.4	12	35	551	353				
6	0.1	0	0.1	0.2	0.4						
	0.1	**0**.2	**0**.3	**1**.6	**24**						
	0.1	5.0	0.6	5.9	205						
7	0.1	0.1	0.2	0.4							
	0.1	**0**.3	**0**.7	**4**.1							
	0.2	5.4	4.6	22							
8	0.2	0.2	0.2	0.3							
	0.2	**0**.3	**1**.0	**11**							
	0.2	4.8	4.1	111							
10	0.3	0.4	0.4								
	0.5	**0**.5	**2**.8								
	7.0	3.5	12								
15	1.3	1.4	3.5								
	1.7	**1**.9	**25**								
	5.8	5.1	73								
20	3.8	4.0	4.7								
	5.0	**5**.3	**123**								
	9.4	8.1	352								
30	25	26									
	29	**29**									
	41	32									
40	88	90									
	98	**101**									
	106	152									
50	248	259									
	271	**283**									
	300	429									

[6]This method is implemented in the Maple procedure *integral_via_iterated*, defined in the file `iterated_laurent.mpl`.

5.3. **Integration of a monomial, using Taylor expansion.** Here, we test the implementation of the algorithm described in Section 3.4. This algorithm is based on Corollary 11 .[7]

The running times are shown in Table 8.

TABLE 8. Integration of a random monomial of prescribed degree using Taylor expansion.

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
2	0	0	0	0	0	0	0	0	0	0.9	1.9
	0	**0**	**0**	**0**	**0**	**0**.1	**0**.2	**0**.3	**2**.2	**24**	**68**
	0	0	0	0.1	0.1	0.3	0.6	0.9	13	101	426
3	0	0	0	0	0	0	0.1	0			
	0	**0**	**0**	**0**.1	**2**.0	**8**.1	**39**	**61**			
	0	0	0.1	0.4	12	60	277	512			
4	0	0	0	0							
	0	**0**	**0**.1	**1**.5							
	0	0	0.4	34							
5	0	0	0	0.1							
	0	**0**	**0**.1	**4**.9							
	0	0	0.6	48							
6	0	0	0	0.1							
	0	**0**	**0**.4	**29**							
	0	0	1.7	236							
7	0	0	0								
	0	**0**	**0**.5								
	0	0	1.7								
8	0	0	0								
	0	**0**	**1**.1								
	0	0	16								
10	0	0	0								
	0	**0**	**3**.0								
	0	0.1	33								
15	0	0	0.1								
	0	**0**.1	**20**								
	0	0.1	64								
20	0	0	1.2								
	0	**0**.7	**81**								
	0	28	205								
30	0	0									
	0	**0**.7									
	0.1	15									
40	0	0									
	0.1	**1**.2									
	0.1	23									
50	0.1	0.1									
	0.1	**1**.7									
	0.2	18									

[7]This method is implemented in the Maple procedure *integral_via_duality*, defined in the file `duality.mpl`.

5.4. **Integration of a dense homogeneous polynomial over a simplex.** Following the tests on single monomials, we ran tests on random polynomials of varying density. We generated these polynomials using the Maple function `randpoly`, requesting a number r of monomials and the homogeneous degree M. For each monomial, the exponent vector was drawn uniformly from $\{\, \mathbf{M} \in \mathbb{N}^d : |\mathbf{M}| = M \,\}$, and the coefficient is drawn uniformly from $\{1, \ldots, 100\}$. Due to collisions, the generated polynomial actually can have fewer monomials than r.[8]

We only include the results for a family of randomly generated, very dense homogeneous polynomials, where we draw $r = \binom{M+d-1}{d-1}$ random monomials. Table 9 shows the number of monomials in the resulting polynomials for our random tests. Tables 10, 12, 11 show the test results of the three methods.

We remark that in the case of the method using decompositions into powers of linear forms, we note that the same powers of linear forms appear in the decomposition formulas (13) for many different monomials $\mathbf{x}^{\mathbf{M}_1}$, $\mathbf{x}^{\mathbf{M}_2}$. We take advantage of this fact by collecting the coefficients of powers of linear forms.[9]

5.5. **Integration of a monomial with few effective variables.** Finally, we tested the performance of the three algorithms on monomials $\mathbf{x}^{\mathbf{M}}$ with a small number D of effective variables. We fix the number D. Then, for a given dimension $n \geq D$ and total degree M, we picked 50 combinations of a random simplex Δ of dimension n and a random exponent vector $\mathbf{M} = (M_1, \ldots, M_D, 0, \ldots, 0)$ with $|\mathbf{M}| = M$. We only include the results for $D = 2$ in Tables 13, 15, and 14.

[8]This is implemented in the Maple procedure `random_sparse_homogeneous_polynomial_with_degree` in the file `examples.mpl`.

[9]This is implemented in the procedure `list_integral_via_waring`.

TABLE 9. Number of monomials in a test family of random dense homogeneous polynomials of prescribed degree

	Degree											
n	1	2	5	10	20	30	40	50	100	200	300	1000
1	1	1	1	1	1	1	1	1	1	1	1	1
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	1	1	1	1	1	1	1	1	1	1	1	1
2	2	1	2	4	11	16	22	28	59	120	179	606
	2	**2**.2	**4**.2	**7**.1	**14**	**20**	**26**	**33**	**64**	**128**	**189**	**634**
	2	3	6	9	18	25	33	37	70	140	198	659
3	3	2	11	37	133	299	524	813	3209	12755		
	3	**4**	**13**	**42**	**146**	**313**	**543**	**839**	**3262**	**12836**		
	3	6	17	46	160	325	560	867	3307	12923		
4	4	4	31	168	1085	3396	7748	14719				
	4	**6**.5	**35**	**182**	**1118**	**3452**	**7802**	**14807**				
	4	8	40	196	1144	3499	7887	14914				
5	5	8	72	610	6650	29181						
	5	**9**.7	**81**	**633**	**6716**	**29329**						
	5	12	88	661	6789	29536						
6	6	10	146	1847	33406							
	6	**14**	**160**	**1896**	**33591**							
	6	16	174	1934	33880							
7	7	15	276	5015								
	7	**18**	**291**	**5062**								
	7	21	304	5114								
8	8	19	476	12213								
	8	**23**	**498**	**12295**								
	8	29	521	12396								
10	10	29	1236									
	10	**36**	**1262**									
	10	41	1289									
15	15	64	7274									
	15	**76**	**7352**									
	15	85	7442									
20	20	123	26742									
	20	**133**	**26880**									
	20	146	26985									
30	30	282										
	30	**295**										
	30	309										
40	40	502										
	40	**521**										
	40	542										

TABLE 10. Integration of a random dense homogeneous polynomial of prescribed degree by decomposition into a sum of powers of linear forms

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
2	0	0	0	0.1	1.0	3.1	7.8	16	228		
	0	**0**	**0**	**0**.1	**1**.2	**3**.6	**9**.0	**18**	**270**		
	0	0	0.1	0.2	1.4	4.4	11	21	304		
3	0	0	0.1	2.1	74						
	0	**0**	**0**.2	**2**.5	**81**						
	0.1	0.1	0.3	2.9	88						
4	0	0	0.6	30							
	0	**0**.1	**0**.8	**32**							
	0.1	0.1	1.0	35							
5	0	0.1	2.6								
	0	**0**.1	**3**.0								
	0.2	0.2	3.4								
6	0	0.1	8.9								
	0.1	**0**.2	**9**.5								
	0.2	0.3	10								
7	0.1	0.2	26								
	0.1	**0**.3	**28**								
	0.2	0.4	30								
8	0.1	0.4	81								
	0.1	**0**.5	**85**								
	0.2	0.7	91								
10	0.2	0.9									
	0.3	**1**.0									
	0.4	1.2									
15	0.6	4.0									
	0.7	**4**.3									
	0.8	4.8									
20	1.6	12									
	1.8	**13**									
	1.9	14									
30	5.8	63									
	6.1	**68**									
	6.4	72									
40	15	221									
	16	**232**									
	16	241									
50	29										
	30										
	30										

TABLE 11. Integration of a random dense homogeneous polynomial of prescribed degree using iterated Laurent series

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
2	0	0	0	0	0.2	0.6	1.2	2.1	24	255	
	0	**0**	**0**	**0**.1	**0**.3	**0**.8	**1**.7	**3**.1	**29**	**350**	
	0.1	0.1	0.1	0.2	0.4	1.0	2.2	4.0	37	438	
3	0	0	0.2	2.1	22	146					
	0	**0**.1	**0**.4	**2**.7	**32**	**184**					
	0.2	0.2	0.6	3.3	37	210					
4	0	0.1	2.4	37							
	0.1	**0**.2	**3**.0	**40**							
	0.2	0.3	3.6	44							
5	0.2	0.3	12	351							
	0.2	**0**.5	**14**	**365**							
	0.3	0.7	16	380							
6	0.3	1.0	50								
	0.4	**1**.2	**55**								
	0.5	1.5	61								
7	0.7	2.0	161								
	0.8	**2**.4	**179**								
	0.9	2.8	195								
8	1.2	3.3	481								
	1.3	**4**.6	**517**								
	1.6	5.8	560								
10	3.5	13									
	3.6	**15**									
	3.8	18									
15	22	117									
	23	**127**									
	23	140									
20	91										
	94										
	98										

TABLE 12. Integration of a random dense homogeneous polynomial of prescribed degree using Taylor expansion.

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
2	0	0	0	0	0.1	0.3	1.2	1.2	15		
	0	**0**	**0**	**0**.1	**0**.2	**0**.8	**2**.3	**5**.6	**98**		
	0	0	0.1	0.1	0.3	1.1	3.1	7.4	129		
3	0	0	0.1	1.1	60						
	0	**0**	**0**.3	**2**.4	**132**						
	0	0	2.0	3.3	158						
4	0	0	1.1	62							
	0	**0**	**1**.6	**117**							
	0	0.1	2.6	143							
5	0	0.1	4.4								
	0	**0**.1	**8**.1								
	0	0.1	24								
6	0	0.1	28								
	0	**0**.2	**35**								
	0.1	2.3	41								
7	0	0.2	117								
	0	**0**.4	**136**								
	0.1	5.5	150								
8	0	0.4	366								
	0.1	**0**.7	**431**								
	0.1	12	475								
10	0.1	0.8									
	0.1	**1**.3									
	0.1	9.8									
15	0.1	4.0									
	0.2	**5**.6									
	0.2	7.3									
20	0.3	16									
	0.4	**18**									
	2.8	20									
30	0.9	95									
	1.1	**104**									
	4.3	109									
40	2.3	370									
	2.6	**384**									
	4.4	403									
50	5.2										
	5.3										
	5.5										

TABLE 13. Integration of a monomial of prescribed degree with 2 effective variables by decomposition into a sum of powers of linear forms

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
3	0	0	0	0	0	0	0	0	1.9	1.9	9.8
	0	**0**	**0**	**0**.1	**0**.3	**0**.7	**1**.5	**2**.2	**11**	**47**	**109**
	0	0.2	0.1	0.2	0.5	1.1	2.3	3.5	15	80	217
4	0	0	0	0	0	0	0	0.5	0.1	0.2	0.3
	0	**0**	**0**	**0**.1	**0**.4	**0**.9	**1**.7	**2**.8	**13**	**64**	**180**
	0	0.3	0.2	0.2	0.7	1.4	2.8	4.3	20	103	270
5	0	0	0	0	0	0	0	0	1.8	0.3	6.0
	0	**0**	**0**.1	**0**.1	**0**.6	**1**.3	**2**.5	**3**.5	**17**	**81**	**205**
	0	0.4	0.2	0.3	1.0	2.0	3.8	5.9	26	128	335
6	0	0	0	0	0	0	0	0	2.1	7.0	0.3
	0	**0**	**0**.1	**0**.2	**0**.7	**1**.8	**3**.1	**4**.5	**25**	**105**	**241**
	0.2	0.2	0.2	0.4	1.2	2.6	4.9	7.5	34	164	449
7	0	0	0	0	0	0	0	0	2.3	0.2	0.2
	0	**0**	**0**.1	**0**.2	**1**.0	**1**.9	**3**.7	**5**.9	**24**	**128**	**279**
	0	0.3	0.3	0.5	1.5	3.0	5.9	8.7	39	184	468
8	0	0	0	0	0	0	0	0.1	4.2	0.4	10.0
	0	**0**	**0**.1	**0**.3	**1**.1	**2**.3	**4**.0	**6**.8	**28**	**122**	**347**
	0	0.5	0.3	0.6	1.6	3.5	6.5	10.0	45	206	529
10	0	0	0	0	0	0	0	1.7	0.1	13	
	0	**0**	**0**.1	**0**.4	**1**.3	**3**.3	**5**.6	**10**	**41**	**189**	
	0.3	0.3	0.4	0.7	2.2	4.9	8.6	14	58	266	
15	0	0	0	0	0.1	0.1	0.1	0.1	0.2	0.2	
	0.1	**0**.1	**0**.2	**0**.7	**2**.4	**6**.0	**9**.9	**15**	**65**	**292**	
	0.3	0.3	0.5	1.2	4.0	8.9	16	24	105	479	
20	0	0.1	0	0.1	0.1	0.1	0.1	0.2	23		
	0.1	**0**.2	**0**.5	**1**.3	**4**.7	**9**.7	**17**	**26**	**123**		
	0.3	0.5	0.8	2.1	7.1	15	26	40	170		
30	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.3		
	0.2	**0**.4	**0**.9	**2**.8	**11**	**19**	**38**	**55**	**254**		
	0.4	0.8	1.5	4.7	15	33	58	92	369		
40	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4			
	0.4	**0**.6	**1**.9	**5**.4	**20**	**41**	**66**	**101**			
	0.6	1.2	2.9	8.5	29	62	110	171			
50	0.5	0.5	0.5	0.6	0.6	0.6	0.7	36			
	0.6	**1**.0	**3**.4	**9**.2	**31**	**63**	**130**	**201**			
	0.8	2.1	4.7	14	49	106	185	286			

TABLE 14. Integration of a monomial of prescribed degree with 2 effective variables using iterated Laurent expansion

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
3	0	0	0	0	0	0	0	0	0.2	0.3	0.9
	0	**0**	**0**	**0**	**0**.1	**0**.2	**0**.3	**0**.3	**1**.2	**6**.2	**16**
	0	0	0	0.1	0.1	0.3	5.6	2.1	3.6	20	62
4	0	0	0	0	0.1	0.1	0.1	0.1	0.2	0.4	0.9
	0	**0**	**0**.1	**0**.1	**0**.1	**0**.3	**0**.3	**0**.5	**2**.1	**11**	**47**
	0.1	0.1	0.1	0.1	0.2	4.2	2.5	3.0	5.4	38	166
5	0.1	0	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.6	1.3
	0.1	**0**.1	**0**.1	**0**.1	**0**.3	**0**.3	**0**.5	**0**.7	**3**.6	**21**	**86**
	0.1	0.1	0.1	0.2	6.4	2.3	2.8	3.7	9.8	80	271
6	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.4	1.0	1.9
	0.1	**0**.1	**0**.1	**0**.2	**0**.3	**0**.6	**0**.8	**1**.1	**7**.2	**39**	**100**
	0.2	0.1	0.2	3.4	2.5	2.6	3.8	4.7	30	269	550
7	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.6	1.2	
	0.1	**0**.1	**0**.2	**0**.3	**0**.6	**0**.8	**1**.2	**2**.0	**6**.8	**94**	
	0.2	0.2	3.9	2.5	2.8	3.4	4.2	6.7	26	260	
8	0.2	0.2	0.2	0.2	0.2	0.2	0.3	0.3	0.7	2.0	
	0.2	**0**.3	**0**.3	**0**.4	**0**.7	**1**.2	**1**.9	**2**.9	**16**	**99**	
	0.3	4.5	2.3	2.6	3.1	3.6	5.8	12	42	322	
10	0.3	0.3	0.3	0.4	0.4	0.5	0.6	0.6	1.1		
	0.5	**0**.4	**0**.5	**0**.7	**1**.2	**2**.4	**3**.8	**5**.9	**23**		
	4.2	2.3	2.5	3.0	3.9	6.5	11	17	77		
15	1.3	1.3	1.4	1.4	1.6	1.7	1.9	2.4			
	1.4	**1**.5	**1**.7	**2**.1	**4**.0	**7**.3	**11**	**20**			
	1.6	1.9	2.0	3.4	9.1	21	31	52			
20	4.0	4.0	4.1	4.2							
	4.2	**4**.4	**4**.8	**6**.1							
	4.4	5.6	5.7	9.3							
30	22	24	24	25	26						
	25	**25**	**27**	**32**	**47**						
	26	28	31	45	96						
40	71	88	90								
	85	**92**	**95**								
	92	102	103								
50	196	228	212	217							
	231	**239**	**244**	**261**							
	252	249	277	347							

TABLE 15. Integration of a monomial of prescribed degree with 2 effective variables using Taylor expansion

	Degree										
n	1	2	5	10	20	30	40	50	100	200	300
3	0	0	0	0	0	0	0	0	0	1.9	
	0	**0**	**0**	**0**	**0**.1	**0**.2	**0**.4	**0**.8	**5**.8	**62**	
	0	0	0.1	0.1	0.2	0.6	1.3	2.6	27	359	
4	0	0	0	0	0	0	0	0	0		
	0	**0**	**0**	**0**	**0**.2	**0**.4	**0**.7	**1**.8	**11**		
	0	0	0.1	0.1	0.4	1.2	3.0	8.2	58		
5	0	0	0	0	0	0	0	0	0		
	0	**0**	**0**	**0**.1	**0**.2	**0**.6	**2**.0	**2**.9	**19**		
	0	0	0.1	0.2	0.7	2.2	6.7	15	110		
6	0	0	0	0	0	0	0	0	0		
	0	**0**	**0**	**0**.1	**0**.7	**0**.9	**2**.0	**4**.4	**31**		
	0	0	0.1	0.3	16	8.4	8.8	72	124		
7	0	0	0	0	0	0	0.2	0	0		
	0	**0**	**0**	**0**.1	**0**.5	**1**.6	**4**.3	**5**.9	**52**		
	0	0.1	0.1	0.3	1.8	8.4	21	32	246		
8	0	0	0	0	0	0	0	0.3	0.1		
	0	**0**	**0**.1	**0**.2	**0**.9	**2**.7	**3**.6	**9**.3	**65**		
	0.1	0.1	0.2	0.4	2.8	36	19	42	547		
10	0	0	0	0	0	0	0	0			
	0	**0**	**0**.1	**0**.2	**1**.1	**3**.5	**8**.4	**20**			
	0.1	0.1	0.2	0.6	7.0	22	40	92			
20	0	0	0	0	0	0	0	2.0			
	0	**0**.1	**0**.4	**2**.8	**12**	**40**	**74**	**121**			
	0.1	0.2	1.1	9.3	43	156	309	462			
30	0	0	0	0	0	0.1					
	0	**0**.2	**1**.4	**6**.9	**62**	**90**					
	0.1	0.5	5.6	29	205	583					
40	0	0	0.1	0.1	0.1						
	0.1	**0**.4	**3**.3	**10**	**71**						
	0.1	0.9	32	40	329						
50	0.1	0.1	0.1	0.1	0.1						
	0.1	**0**.4	**5**.5	**19**	**126**						
	0.1	1.5	39	65	525						

5.6. **Discussion.** In our implementation of the three methods and our experiments for the case of random monomials, we observe that the method of iterated Laurent expansion is faster than the two other methods if the dimension n is very small (up to $n = 5$). Starting from dimension $n = 6$, the method using decompositions into powers of linear forms is faster than the other two methods. The method using Taylor expansion is always inferior to the better of the two other methods, for any combination of degree and dimension.

In the experiments with random dense polynomials, in our implementation we did not see significant savings from collecting the coefficients of the same powers of linear forms. As a consequence, the ranking of the three methods is the same as it is in the case of random monomials.

The experiments with random monomials with few effective variables show that all three methods benefit from using few effective variables. The greatest effect is on the method using decompositions into powers of linear forms, where, for example, the restriction to 2 effective variables allows to handle combinations of high degree $M = 200$ and high dimension $n = 15$. However, for low dimensions ($n \leq 5$), the method of iterated Laurent expansion still wins. Also here the method using Taylor expansion is always inferior to the better of the two other methods. This discussion shows the power of Brion's formula.

6. Conclusions

We discussed various algorithms for the exact integration of polynomials over simplicial regions. Beside their theoretical efficiency, the simple rough experiments we performed clearly demonstrated that these methods are robust enough to attack rather difficult problems. Our investigations opened several doors for further development, which we will present in a forthcoming paper.

First, we have some theoretical issues expanding on our results. As in the case of volumes and the computation of centroids, it is likely that our hardness result, Theorem 1, can be extended into an inapproximability result as those obtained in [28]. Another goal is to study other families of polytopes for which exact integration can be done efficiently. Furthermore, we will present a natural extension of the computation of integrals, the efficient computation of the highest degree coefficients of a weighted Ehrhart quasipolynomial of a simplex. Besides the methods of the present article, these last computations are based on the results of [7] and [6].

Second, our intention has been all along to develop algorithms with a good chance of becoming practical and that allow for clear implementation. Thus we have also some practical improvements to discuss.

Finally, in order to develop practical integration software, it appears that our methods should be coupled with fast techniques for decomposing domains into polyhedral regions (e.g. triangulations).

7. Acknowledgements

A part of the work for this paper was done while the five authors visited Centro di Ricerca Matematica Ennio De Giorgi at the Scuola Normale Superiore of Pisa, Italy. We are grateful for the hospitality of CRM and the support we received. The third author was also supported by NSF grant DMS-0608785. We are grateful for help from Christophe Margerin and comments from Jean-Bernard Lasserre and Bernd Sturmfels.

References

[1] J. Alexander and A. Hirschowitz, *Polynomial interpolation in several variables*, J. Algebraic Geom. **4** (1995), 201–222.

[2] V. Baldoni, N. Berline, J. A. De Loera, M. Köppe, and M. Vergne, *Maple programs accompanying the manuscript* How to integrate a polynomial over a simplex, http://www.math.ucdavis.edu/~mkoeppe/art/pisa-integration-experiments/, 2008.

[3] A. I. Barvinok, *Computation of exponential integrals*, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI) Teor. Slozhn. Vychisl. **5** (1991), 149–162, 175–176, translation in J. Math. Sci. 70 (1994), no. 4, 1934–1943.

[4] ———, *Exponential integrals and sums over convex polyhedra (in Russian)*, Funktsional. Anal. i Prilozhen. **26** (1992), no. 2, 64–66, translated in Funct. Anal. Appl. 26 (1992), no. 2, pp. 127–129.

[5] ———, *Partition functions in optimization and computational problems*, Algebra i Analiz **4** (1992), 3–53, translation in St. Petersburg Math. J. 4 (1993), no. 1, pp. 1–49.

[6] N. Berline and M. Vergne, *Local Euler–Maclaurin formula for polytopes*, Moscow Math. J. **7** (2007), 355–386.

[7] ———, *Local Euler–Maclaurin expansion of Barvinok valuations and Ehrhart coefficients of rational polytopes*, Contemporary Mathematics **452** (2008), 15–33.

[8] N. Berline, E. Getzler, and M. Vergne, *Heat kernels and Dirac operators*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 298, Springer-Verlag, Berlin, 1992.

[9] A. T. Bharucha-Reid and M. Sambandham, *Random polynomials*, Academic Press, Orlando, Florida, 1986.

[10] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas, *Symmetric tensor decomposition*, arXiv: 0901.3706v2 [cs.SC], 2009.

[11] M. C. Brambilla and G. Ottaviani, *On the Alexander–Hirschowitz theorem*, e-print arXiv:math.AG/0701409v2, 2007.

[12] G. Brightwell and P. Winkler, *Counting linear extensions*, Order **8** (1991), no. 3, 225–242.

[13] M. Brion, *Points entiers dans les polyédres convexes*, Ann. Sci. École Norm. Sup. **21** (1988), no. 4, 653–663.

[14] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic complexity theory*, Grundlehren der mathematischen Wissenschaften, no. 315, Springer, Berlin, 1997.

[15] J. A. De Loera, J. Rambau, and F. Santos, *Triangulations: Structures and algorithms*, Book manuscript, 2008.

[16] M. E. Dyer and A. M. Frieze, *On the complexity of computing the volume of a polyhedron*, SIAM J. Comput. **17** (1988), no. 5, 967–974.

[17] G. Elekes, *A geometric inequality and the complexity of computing volume*, Discrete Comput. Geom. **1** (1986), no. 4, 289–292.

[18] M. R. Garey and D. S. Johnson, *Computers and intractibility: A guide to the theory of NP-completeness*, W. H. Freeman and Co., San Francisco, California, 1979.

[19] R. Kannan and A. Bachem, *Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix*, SIAM J. Comput. **8** (1979), no. 4, 499–507.

[20] L. Khachiyan, *Complexity of polytope volume computation*, New trends in discrete and computational geometry, Algorithms Combin., vol. 10, Springer, Berlin, 1993, pp. 91–101.
[21] J. B. Lasserre, *Integration on a convex polytope*, Proceedings of the AMS **126** (1998), no. 8, 2433–2441.
[22] J. B. Lasserre and K. Avrachenkov, *The multidimensional version of* $\int_a^b x^p \, \mathrm{d}x$, American Mathematical Monthly **108** (2001), no. 2, 151–154.
[23] M. Laurent, *Sums of squares, moment matrices, and optimization over polynomials*, In "Emerging applications of algebraic geometry (Minneapolis, Minnesota) (M. Putinar and S. Sullivant, eds.), IMA volumes in Mathematics and its Applications, 2008.
[24] J. Lawrence, *Polytope volume computation*, Math. Comp. **57** (1991), no. 195, 259–271.
[25] S. Lin, B. Sturmfels, and Z. Xu, *Marginal likelihood integrals for mixtures of independence models*, e-print arXiv:0805.3602 [stat.CO], 2008.
[26] G. Matera, *Integration of multivariate rational functions given by straight-line programs*, Proceedings of the 11th international symposium on applied algebra, algebraic algorithms and error-correcting codes (London), Lecture notes in Computer Science, vol. 948, Springer, 1995, pp. 347–364.
[27] T. S. Motzkin and E. G. Straus, *Maxima for graphs and a new proof of a theorem of Turán*, Canadian Journal of Mathematics **17** (1965), 533–540.
[28] L. Rademacher, *Approximating the centroid is hard*, Proceedings of 23th annual ACM Symposium of Computational Geometry, Gyeongju, South Korea, June 6-8, 2007, 2007, pp. 302–305.
[29] M. Schechter, *Integration over a polyhedron: An application of the Fourier–Motzkin elimination method*, American Mathematical Monthly **105** (1998), no. 3, 246–251.
[30] G. M. Ziegler, *Lectures on polytopes*, Graduate texts in Mathematics, no. 152, Springer, New York, 1995.
[31] O. C. Zienkiewicz and R. L. Taylor, *The finite element method*, McGraw-Hill, London, 1988.

Velleda Baldoni: Dipartimento di Matematica, Università degli studi di Roma "Tor Vergata", Via della ricerca scientifica 1, I-00133, Italy
E-mail address: baldoni@mat.uniroma2.it

Nicole Berline: Centre de Mathématiques Laurent Schwartz, École Polytechnique, 91128 Palaiseau Cedex, France
E-mail address: nicole.berline@math.polytechnique.fr

Jesús A. De Loera: Department of Mathematics, University of California, Davis, One Shields Avenue, Davis, CA, 95616, USA
E-mail address: deloera@math.ucdavis.edu

Matthias Köppe: Department of Mathematics, University of California, Davis, One Shields Avenue, Davis, CA, 95616, USA
E-mail address: mkoeppe@math.ucdavis.edu

Michèle Vergne: Centre de Mathématiques Laurent Schwartz, École Polytechnique, 91128 Palaiseau Cedex, France
E-mail address: vergne@math.polytechnique.fr