Vector partition function and representation theory

Charles Cochet

Abstract. We apply some recent developments of Baldoni-Beck-Cochet-Vergne [**BBCV05**] on vector partition function, to Kostant's and Steinberg's formulae, for classical Lie algebras A_r, B_r, C_r, D_r. We therefore get efficient `Maple` programs that compute for these Lie algebras: the multiplicity of a weight in an irreducible finite-dimensional representation; the decomposition coefficients of the tensor product of two irreducible finite-dimensional representations. These programs can also calculate associated Ehrhart quasipolynomials.

Nous appliquons des résultats récents de Baldoni-Beck-Cochet-Vergne [**BBCV05**] sur la fonction de partition vectorielle, aux formules de Kostant et de Steinberg, dans le cas des algèbres de Lie classiques A_r, B_r, C_r, D_r. Ceci donne lieu à des programmes `Maple` efficaces qui calculent pour ces algèbres de Lie : la multiplicité d'un poids dans une représentation irréductible de dimension finie ; les coefficients de décomposition du produit tensoriel de deux représentations irréductibles de dimension finie. Ces programmes permettent également d'évaluer les quasipolynômes d'Ehrhart associés.

1. Introduction

In this note, we are interested in the two following computational problems for classical Lie algebras A_r, B_r, C_r, D_r:

- The multiplicity c_λ^μ of the weight μ in the representation $V(\lambda)$ of highest weight λ.
- Littlewood-Richardson coefficients, that is the multiplicity $c_{\lambda\ \mu}^{\ \nu}$ of the representation $V(\nu)$ in the tensor product of representations of highest weights λ and μ.

Softwares LiE (from van Leeuwen *et al.* [**vL94**]) and GAP [**GAP**]), and `Maple` packages `coxeter/weyl` (from Stembridge [**S95**]), use Freudenthal's and Klimyk's formulae, and work for any semi-simple Lie algebra (not only for classical Lie algebras). Unfortunately, these formulae are really sensitive to the size of coefficients of weights. Moreover, they do not lead to the computation of associated quasipolynomials $(\lambda, \mu) \mapsto c_\lambda^\mu$ and $(\lambda, \mu, \nu) \mapsto c_{\lambda\ \mu}^{\ \nu}$.

Here the approach to these two problems is through *vector partition function*, that is the function computing the number of ways one can decompose a vector as a linear combination with nonnegative integral coefficients of a fixed set of vectors.

2000 *Mathematics Subject Classification.* Primary 22E46, 52B55.

Key words and phrases. weight multiplicity, Littlewood-Richardson coefficients, Kostka numbers, Ehrhart quasipolynomial, convex polytopes, partition function.

For example the number $p(x)$ of ways of counting x euros with coins, that is

$$p(x) = \sharp\{n \in \mathbb{Z}_+^8\,;\, x = 200n_1 + 100n_2 + 50n_3 + 20n_4 + 10n_5 + 5n_6 + 2n_7 + n_8\},$$

can be seen as the partition of the 1-dimensional vector (x) with respects to the set $\{(200), (100), (50), (20), (10), (5), (2), (1)\}$ of 1-dimensional vectors. In the case of the decomposition with respects to the set of positive roots of a simple Lie algebra, we speak of *Kostant partition function*.

Recall that any d-dimensional rational convex polytope can be written as the set $P(\Phi, a)$ of nonnegative solutions $x = (x_i) \in \mathbb{R}^N$ of an equation $\sum_{i=1}^N x_i\phi_i = a$, for a matrix Φ with columns $\phi_i \in \mathbb{Z}^r$ and $a \in \mathbb{Z}^r$ $(d = N - r)$. It follows that evaluating the vector partition is equivalent to computing the number of integral points in a rational convex polytope.

The vector partition function arises in many areas of mathematics: representation theory, flows in networks, magic squares, statistics, crystal bases of quantum groups. Its complexity is polynomial in the size of input when the dimension of the polytope is fixed, and NP-hard if it can vary [**B94, B97, BP99**].

There are several approaches to the vector partition problem. For example Barvinok's decomposition algorithm [**B94**], recently implemented by the `LattE` team [**DHTY03, L**], works for general sets of vectors. Beck-Pixton [**BP03**] also created an algorithm dedicated to the vector set arising from the Birkhoff polytope, counting the number of semi-magic squares.

In this note, we use recent results of Baldoni-Beck-Cochet-Vergne [**BBCV05**] to obtain a fast algorithm for Kostant partition function *via* inverse Laplace formula. These results involve DeConcini-Procesi's *maximal nested sets* (or in short MNSs [**DCP04**]) and iterated residues of rational functions computed by formal power series development.

We combine resulting procedures with Kostant's and Steinberg's formulae giving c_λ^μ and ${c_\lambda}^\nu{}_\mu$ in terms of vector partition function. We then obtain a `Maple` program computing for classical Lie algebras (A_r, B_r, C_r, D_r), the multiplicity of a weight in an irreducible finite-dimensional representation, as well as decomposition coefficients of the tensor product of two irreducible finite-dimensional representations. To the best of our knowledge, they are also the only ones able to compute associated piecewise-defined quasipolynomials $(\lambda, \mu) \mapsto c_\lambda^\mu$ and $(\lambda, \mu, \nu) \mapsto {c_\lambda}^\nu{}_\mu$.

These programs (available at [**C**]) are specially designed for large parameters of weights. Indeed although only written in `Maple` they can perform examples with weights with 5 digits coordinates, far beyond classical softwares written in `C++`. We also stress that our programs are absolutely clear, easy to use and require no installation of exotic package or program. Retro-compatibility has been checked downto `Maple Vr5`. They are fully commented, so that a curious user can figure out their internal mechanisms.

However, certain other softwares and packages are not limited by the rank of the algebra like our programs. For example computation of non-trivial examples in Lie algebras of rank 10 is possible with the software LiE, whereas our programs are efficient up to rank 5–7. These facts make our programs complementary to traditional softwares.

Remark that Kostant's and Steinberg's formulae have already been implemented once in the case of A_r [**C03**]. This previous program relies on results of Baldoni-Vergne [**BV01**] implemented by Baldoni-DeLoera-Vergne [**BdLV03**], computing Kostant partition function only in the case of A_r. Tools were *special permutations* and again iterated residues of rational fraction.

A new technique for Littlewood-Richardson coefficients has been recently designed by DeLoera-McAllister [**DM05**]. For A_r, they wrote an algorithm using hive polytopes [**KT99**]. For B_r, C_r, D_r, they implemented Berenstein-Zelevinsky polytopes [**BZ01**]. They can also evaluate stretched Littlewood-Richardson coefficients ${c_{t\lambda}}^{t\nu}{}_{t\mu}$. These two methods consist in computing a tensor product coefficient as the number of lattice points in just one specific convex rational polytope. However our programs based on multidimensional residues are faster, and can reach examples not available by their method.

This paper is organized as follows. Section 2 recalls representation theory problems we are interested in and links them with algebraic combinatorics. Section 3 describes more precisely rational convex polytopes and formulae counting their integral points. Section 4 introduces maximal nested sets and formulae that were used in our programs. Finally in Section 5 we perform tests of our programs.

2. Representation theory and convex polytopes

Let us fix the notations once and for all. Let $\mathfrak{g}$ be a semi-simple Lie algebra of rank r. Choose a Cartan subalgebra $\mathfrak{t}$ of $\mathfrak{g}$ and denote by $L \subset \mathfrak{t}^*$ the weight lattice.

Let Δ^+ be a positive roots system. The root lattice is defined as $\mathbb{Z}[\Delta^+]$. Let $C(\Delta^+)$ be the cone spanned by linear combinations with nonnegative coefficients of positive roots. The Weyl group of $\mathfrak{g}$ for $\mathfrak{t}$ is denoted by W.

There exist only four simple Lie algebras A_r, B_r, C_r, D_r of rank r, called *classical Lie algebras* of rank r [**Bou68**], and determined by their positive roots systems:

$$\begin{array}{ll}
A_r: & \Delta^+ = \{e_i - e_j \,|\, 1 \leq i < j \leq r+1\} \subset \mathbb{R}^{r+1}, \\
B_r: & \Delta^+ = \{e_i - e_j \,|\, 1 \leq i < j \leq r\} \cup \{e_i \,|\, 1 \leq i \leq r\} \subset \mathbb{R}^{r}, \\
C_r: & \Delta^+ = \{e_i - e_j \,|\, 1 \leq i < j \leq r\} \cup \{2e_i \,|\, 1 \leq i \leq r\} \subset \mathbb{R}^{r}, \\
D_r: & \Delta^+ = \{e_i - e_j \,|\, 1 \leq i < j \leq r\} \cup \{e_i + e_j \,|\, 1 \leq i < j \leq r\} \subset \mathbb{R}^{r}.
\end{array}$$

The character of a representation V of $\mathfrak{g}$ is $\mathrm{ch}(V) = \sum_{\mu \in L} \dim(V_\mu) e^\mu$. Recall that the irreducible finite-dimensional representation of $\mathfrak{g}$ of highest weight λ is denoted by $V(\lambda)$. Hence the weight multiplicity c_λ^μ is defined as $\dim(V(\lambda)_\mu)$ for any weight μ such that $\lambda - \mu$ is in the root lattice. Multiplicities c_λ^μ are called *Kostka numbers* when $\mathfrak{g} = A_r = \mathfrak{sl}_{r+1}(\mathbb{C})$.

On the other hand, multiplicities of representations $V(\nu)$ in the tensor product $V(\lambda) \otimes V(\mu)$ are called *Littlewood-Richardson coefficients* (or *Clebsch-Gordan coefficients*). Here ν is a dominant weight such that $\lambda + \mu - \nu$ is in the root lattice.

Evaluating weight multiplicities and Littlewood-Richardson coefficients is a difficult task. For A_1, computing Kostka numbers is immediate and Clebsch-Gordan's formula gives Littlewood-Richardson coefficients. For A_2, one can still compute some small examples. But for general X_r ($r \geq 3$) or for weights which components are big (say, with two digits), direct computation is usually intractable.

There exist many formulae from representation theory for c_λ^μ and ${c_\lambda}^\nu{}_\mu$. The first one, valid in any complex semi-simple Lie algebra $\mathfrak{g}$, is Weyl's character formula

$$\mathrm{ch}(V(\lambda)) \quad = \quad \frac{A_{\lambda+\rho}}{A_\rho}, \qquad \text{where } A_\mu = \sum_{w \in W} (-1)^{\varepsilon(w)} e^{w(\mu)},$$

where ρ is half the sum of positive roots for $\mathfrak{g}$. Littlewood-Richardson coefficients are obtained from this formula, since the character of $V(\lambda)\otimes V(\mu)$ is

$$\mathrm{ch}(V(\lambda)\otimes V(\mu)) \;=\; \mathrm{ch}(V(\lambda))\times \mathrm{ch}(V(\mu)) \;=\; \sum_{\nu\in L\,;\,\lambda+\mu-\nu\in\mathbb{Z}[\Delta^+]} c_{\lambda}{}^{\nu}{}_{\mu}\, \mathrm{ch}(V(\nu)).$$

But these two formulae do not lead to efficient computations when the rank of $\mathfrak{g}$ or the size of coefficients of weights grow. Moreover, computing the whole character is untractable: for $\mathfrak{g} = A_3 = \mathfrak{sl}_4(\mathbb{C})$ and $\lambda = (2, 1, 0, -3)$, the character $\mathrm{ch}(V(\lambda))$ has 9 monomials but the character $\mathrm{ch}(V(10\lambda))$ has 2903 monomials.

Let us describe Kostant's and Steinberg's formulae in the case of any semi-simple Lie algebra $\mathfrak{g}$. Denote by $k_{\mathfrak{g}}(a)$ the number of ways one can write a vector a as a nonnegative linear combination of positive roots. Remark that $k_{\mathfrak{g}}(a) = 0$ unless a is in the root lattice $\mathbb{Z}[\Delta^+]$. This number satisfies the equation

$$\frac{1}{\prod_{\alpha\in\Delta^+}(1-e^{-\alpha})} = \sum_{a\in\mathbb{Z}[\Delta^+]} k_{\mathfrak{g}}(a)\; e^{-a}.$$

Let λ and μ be respectively a dominant weight and a weight such that $\lambda-\mu \in \mathbb{Z}[\Delta^+]$. A Weyl group element $w \in W$ is *valid* for λ and μ if the root lattice element $w(\lambda+\rho)-(\mu+\rho)$ is in the cone $C(\Delta^+)$. The set of such w's is denoted by $\mathrm{Val}(\lambda,\mu)$. Then Kostant's formula asserts that the weight multiplicity c_{λ}^{μ} equals

$$c_{\lambda}^{\mu} = \sum_{w\in\mathrm{Val}(\lambda,\mu)} (-1)^{\varepsilon(w)}\, k_{\mathfrak{g}}(w(\lambda+\rho)-(\mu+\rho)). \tag{2.1}$$

Similarly let λ, μ, ν, be three dominant weights such that $\lambda+\mu-\nu \in \mathbb{Z}[\Delta^+]$. The couple $(w, w') \in W\times W$ is *valid* for λ, μ, ν, if the root lattice element $w(\lambda+\rho)+w'(\mu+\rho)-(\nu+2\rho)$ is in $C(\Delta^+)$. The set of such couples is denoted by $\mathrm{Val}(\lambda,\mu,\nu)$. Then Steinberg's formula asserts that the Littlewood-Richardson coefficient equals

$$c_{\lambda}{}^{\nu}{}_{\mu} = \sum_{(w,w')\in\mathrm{Val}(\lambda,\mu,\nu)} (-1)^{\varepsilon(w)+\varepsilon(w')} k_{\mathfrak{g}}(w(\lambda+\rho)+w'(\mu+\rho)-(\nu+2\rho)). \tag{2.2}$$

Sets of valid Weyl group elements and valid couples of Weyl group elements turn out to be relatively small, when compared to W and $W\times W$ (which size is exponential in the rank). Remark that Kostant's (resp. Steinberg's) formula also work when $\lambda-\mu$ (resp. $\lambda+\mu-\nu$) is not in the root lattice, since Kostant partition function vanishes on vectors that are not in the root lattice.

From now on, let X_r be a classical Lie algebra of rank r. Here X stands for A, B, C, D. Its positive roots system will be denoted by X_r^+.

Multiplicities c_{λ}^{μ} and $c_{\lambda}{}^{\nu}{}_{\mu}$ behave nicely, in function of the parameters. More precisely, there exists a decomposition of the space $\mathfrak{t}^*\oplus\mathfrak{t}^*\oplus\mathfrak{t}^*$ in union of closed cones C, such that the restriction of $c_{\lambda}{}^{\nu}{}_{\mu}$ to each cone C is given by a quasi-polynomial function. This follows from theorems of Knutson-Tao [**KT99**] (for A_r), Berenstein-Zelevinsky [**BZ01**] (for any semi-simple Lie algebra) giving $c_{\lambda}{}^{\nu}{}_{\mu}$ as the number of points in a rational convex polytope. In the case of A_r, the fact that $c_{\lambda}{}^{\nu}{}_{\mu}$ is given on each cone C by a polynomial function is proven in Rassart [**Ras04**], and the case of A_3 is treated as an illustration. The description of the decomposition of $\mathfrak{t}^* \oplus \mathfrak{t}^*$ in cones C, where the function c_λ^μ is polynomial for A_r, was given for low ranks by Billey-Guillemin-Rassart [**BGR03**]. See also Rassart's website [**R**] for wonderful slides.

The common point to Kostant's and Steinberg's formulae is the function counting the number of decompositions of a root lattice element as a linear combination with nonnegative integral coefficients of positive roots of the Lie algebra. The next section deals with an efficient method to compute it.

3. Counting integral points in rational convex polytopes

3.1. Vector partition function. Let $E \simeq \mathbb{R}^r$ and Φ be an integral matrix with set of columns $\Delta^+ = \{\phi_1, \ldots, \phi_N\} \subset E^*$. Choose $a \in \mathbb{Z}^r$. The rational convex polyhedron associated to Φ and a is

$$P(\Phi, a) \quad = \quad \left\{ x \in \mathbb{R}^N ; \sum_{i=1}^N x_i \phi_i = a,\, x_i \geq 0 \right\}.$$

Remark 3.1. Every convex polyhedron can be realized under the form $P(\Phi, a)$, that is as a set satisfying equality constraints on nonnegative variables. Indeed any inequality can be replaced by an equality by adding a new variable. For example polytopes $\{(x, y) \in \mathbb{R}^2 ; x \geq 0, y \geq 0, x + y \leq 1\}$ and $\{(x, y, z) \in \mathbb{R}^3 ; x \geq 0, y \geq 0, z \geq 0, x + y + z = 1\}$ are isomorphic and have the same number of integral points.

We assume that a is in the cone $C(\Phi)$ spanned by nonnegative linear combinations of the vectors ϕ_i, so that $P(\Phi, a)$ in non-empty. We also assume that the kernel of Φ intersects trivially with the positive orthant $\mathbb{R}^N_+$, so that the cone $C(\Phi)$ is acute and $P(\Phi, a)$ is a polytope (*i.e.* bounded). Finally, we assume that Φ has rank r. The *vector partition function* is by definition

$$k(\Phi, a) \quad = \quad \left| P(\Phi, a) \cap \mathbb{Z}^N_+ \right|,$$

that is the number of nonnegative integral solutions $(x_1, \ldots, x_N)$ of the equation $\sum_{i=1}^N x_i \phi_i = a$. If $\Phi = \Phi(X_r)$ is the matrix which columns are positive roots for a classical Lie algebra X_r, then $a \mapsto k(\Phi(X_r), a)$ is the *Kostant partition function*. For example

$$\Phi(A_2) = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{pmatrix} \quad \text{and} \quad \Phi(B_2) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{pmatrix}.$$

Note that the matrix $\Phi(A_r)$ has rank r (and not $r + 1$), since sums on lines are zero.

A *basic subset* of Δ^+ is a basis $\sigma = \{\alpha_1, \ldots, \alpha_r\}$ of E^* constituted with elements of Δ^+. Let $B(\Delta^+)$ be the collection of all basic subsets of Δ^+. For such a σ, let $C(\sigma)$ be the cone of linear combinations with nonnegative coefficients of α_i's. Denote by $\mathrm{Sing}(\Delta^+)$ the reunion of the facets of cones $C(\sigma)$, $\sigma \in B(\Delta^+)$; this is the set of *singular* vectors. Let $C_{\mathrm{reg}}(\Delta^+) := C(\Delta^+) \setminus \mathrm{Sing}(\Delta^+)$ be the set of *regular* vectors. A *combinatorial chamber* $\mathfrak{c}$ is by definition a connected component of $C_{\mathrm{reg}}(\Delta^+)$. Combinatorial chambers are regions of quasi-polynomiality of the vector partition function $a \mapsto k(\Phi, a)$. Figure 1 represents cones $C(A_3^+)$ and $C(B_3^+)$, and their chamber decompositions.

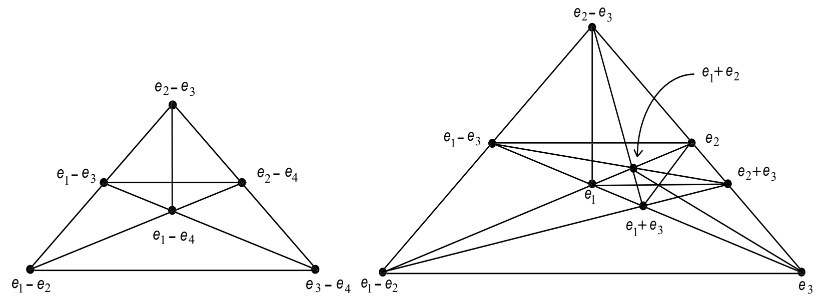

FIGURE 1. The 7 chambers for A_3 and the 23 chambers for B_3

3.2. Brion-Szenes-Vergne formula for classical Lie algebras. Let us describe the formula, computing the number of integral points in rational convex polytopes $P(\Phi(X_r), a)$ associated to a classical algebra X_r, that was implemented in our program.

Let $E = \mathfrak{t}$ and consider the set Δ^+ of positive roots for X_r. Denote by Δ the set $\Delta^+ \cup (-\Delta^+)$ of all roots. Let R_Δ be the vector space of fractions with poles on the hyperplanes defined as kernels of forms $\alpha \in \Delta$. Let S_Δ be the vector space generated by fractions $f_\sigma := \frac{1}{\prod_{\alpha\in\sigma}\alpha}$, $\sigma \in B(\Delta^+)$. Brion-Vergne [**BV97**] proved that R_Δ decomposes as the direct sum $S_\Delta \oplus \partial(R_\Delta)$. We define the *Jeffrey-Kirwan residue* of the chamber $\mathfrak{c}$ as the linear form $\mathrm{JK}_\mathfrak{c}$ on S_Δ:

$$\mathrm{JK}_\mathfrak{c}(f_\sigma) := \begin{cases} \mathrm{vol}(\sigma)^{-1}, & \text{if } \mathfrak{c} \subset C(\sigma), \\ 0, & \text{if } \mathfrak{c} \cap C(\sigma) = \emptyset, \end{cases}$$

where $\mathrm{vol}(\sigma)$ is the volume of the parallelepiped $\sum_{\alpha\in\sigma}[0,1]\alpha$. We extend the JK residue to a linear form on R_Δ by setting it to 0 on $\partial(R_\Delta)$, and to a linear form on the space of formal series $\widehat{R_\Delta}$ by setting it to 0 on homogeneous elements of degree different from $-r$. For example, for the system $\Delta^+ = \{e_1, e_2, e_1+e_2, e_1-e_2\} \subset \mathbb{R}^2$ of positive roots for B_2 and the chamber $\mathfrak{c} = \mathbb{Z}_+e_1 \oplus \mathbb{Z}_+(e_1+e_2)$ we have

$$\mathrm{JK}_\mathfrak{c}\left(\frac{e^{x-y}}{xy^2}\right) = \mathrm{JK}_\mathfrak{c}\left(\frac{x-y}{xy^2}\right) = \mathrm{JK}_\mathfrak{c}\left(\frac{1}{y^2} - \frac{1}{xy}\right) = -1,$$

since $\mathfrak{c} \subset C(\{e_1, e_2\})$.

Let T be the torus $E/E_\mathbb{Z}$, where $E_\mathbb{Z} \subset E$ is the dual of the root lattice. Given a basic subset σ, we define $T(\sigma)$ as the set of elements $g \in T$ such that $e^{\langle\alpha, 2i\pi G\rangle} = 1$ for all $\alpha \in \sigma$; here G is a representative of $g \in E/E_\mathbb{Z}$. Now let

$$\mathcal{F}(g,a)(u) := \frac{e^{\langle a, 2i\pi G+u\rangle}}{\prod_{\alpha\in\Delta}(1 - e^{-\langle\alpha, 2i\pi G+u\rangle})}.$$

THEOREM 3.2 (Brion-Szenes-Vergne [**BV99, SV04**]). *Let $F \subset T$ be a finite set such that $T(\sigma) \subset F$ for all $\sigma \in B(\Delta^+)$. Fix a combinatorial chamber $\mathfrak{c}$. Then for all $a \in \mathbb{Z}[\Delta^+] \cap \bar{\mathfrak{c}}$, we have:*

$$k(\Phi, a) = \sum_{g\in F} \mathrm{JK}_\mathfrak{c}(\mathcal{F}(g,a)).$$

Now that we linked vector partition function and Jeffrey-Kirwan residue, we describe in Section 4 an efficient way to compute the latter.

4. DeConcini-Procesi's maximal nested sets (MNS) [DCP04]

We keep the same notations as in Section 3. A subset $S \subset \Delta^+$ is *complete* if $S = \langle S \rangle \cap \Delta^+$. A complete subset is *reducible* if one can find a decomposition $E = E_1 \oplus E_2$ such that $S = S_1 \cup S_2$ with $S_1 \subset E_1$ and $S_2 \subset E_2$; else S is said *irreducible*. Let $\mathcal{I}$ be the collection of irreducible subsets.

A collection $M = \{I_1, I_2, \ldots, I_s\}$ of irreducible subsets I_j of Δ^+ is *nested*, if: for every subset $\{S_1, \ldots, S_m\}$ of M such that there exist no i, j with $S_i \subset S_j$, the union $S_1 \cup S_2 \cup \cdots \cup S_m$ is complete and the S_i's are its irreducible components. Note that a maximal nested set (MNS in short) has exactly r elements.

Assume Δ^+ irreductible and fix a total order on it. For $M = \{I_1, \ldots, I_s\}$, $I_j \in \Delta^+$, take for every j the maximal element $\beta_j \in I_j$. This defines an application $\phi(M) := \{\beta_1, \ldots, \beta_s\} \subset \Delta^+$. A maximal nested set M is *proper* if $\phi(M)$ is a basis of E^*. Denote by $\mathcal{P}$ the collection of maximal proper nested sets (MPNS in short). We sort $\phi(M)$ and get an ordered list $\theta(M) = [\alpha_1, \ldots, \alpha_r]$. Thus θ is an application from the collection of MPNSs to the collection of ordered basis of E^*. For a given M, let then

$$\begin{aligned} C(M) &:= C(\alpha_1, \ldots, \alpha_r), \\ \mathrm{vol}(M) &:= \mathrm{vol}\left(\oplus_{i=1}^r [0,1]\alpha_i\right), \\ \mathrm{IRes}_M &:= \mathrm{Res}_{\alpha_r=0} \cdots \mathrm{Res}_{\alpha_1=0}\,. \end{aligned}$$

Example 4.1. Let e_i be the canonical basis of $\mathbb{R}^r$, with dual basis e^i ($i = 1, \ldots, r$), and define E as the subspace of vectors which sum of coordinates vanish. Consider the set $\Delta^+ = \{e^i - e^j \,|\, 1 \leq i < j \leq r\}$ of positive roots for A_{r-1}. Irreducible subsets of Δ^+ are indexed by subsets S of $\{1, 2, \ldots, r\}$, the corresponding irreducible subset being $\{e^i - e^j \,|\, i, j \in S, i < j\}$. For instance $S = \{1, 2, 4\}$ parametrizes the set of roots given by $\{e^1 - e^2, e^2 - e^4, e^1 - e^4\}$.

A nested set is represented by a collection $M = \{S_1, S_2, \ldots, S_k\}$ of subsets of $\{1, 2, \ldots, r\}$ such that if S_i, $S_j \in M$ then either $S_i \cap S_j$ is empty, or one of them is contained in another.

For example one can easily compute that for the set of positive roots for A_3 (see Figure 1) there are only 7 MPNS, namely

$$\begin{array}{ll} M_1 = \{[1,2],[1,2,3],[1,2,3,4]\}, & M_2 = \{[2,3],[1,2,3],[1,2,3,4]\}, \\ M_3 = \{[2,3],[2,3,4],[1,2,3,4]\}, & M_4 = \{[3,4],[2,3,4],[1,2,3,4]\}, \\ M_5 = \{[1,3],[2,4],[1,2,3,4]\}, & M_6 = \{[1,2],[3,4],[1,2,3,4]\}. \end{array}$$

Now we can quote the Theorem for the Jeffrey-Kirwan residue computation:

Theorem 4.2 (DeConcini-Procesi). *Let $\mathfrak{c}$ be a combinatorial chamber and fix $f \in R_\Delta$. Take any regular vector $v \in \mathfrak{c}$. Then:*

$$\mathrm{JK}_{\mathfrak{c}}(f) = \sum_{M \in \mathcal{P}\,:\, v \in C(M)} \frac{1}{\mathrm{vol}(M)} \,\mathrm{IRes}_M(f).$$

See [**BBCV05**] for a detailed description of how formulae from Theorems 3.2 and 4.2 were implemented.

5. Our programs

5.1. Description and implementation. Initial data for weight multiplicity and Littlewood-Richardson coefficients are only vectors (respectively two and three). Our programs work with weights represented in the canonical basis of E^*, and not in the fundamental weights basis for X_r. Translation between these two bases is performed *via* straightforward procedures `FromFundaToCanoX(r,v')` and `FromCanoToFundaX(r,v)` (where one replaces `X` by `A`, `B`, `C`, `D`, according to the algebra).

Computation of the weight multiplicity c_λ^μ and of the Littlewood-Richardson coefficient $c_{\lambda}{}^{\nu}{}_{\mu}$ is done by typing in

```
MultiplicityX(lambda,mu);
TensorProductX(lambda,mu,nu);
```

where λ, μ, ν are suitable weights. The syntax for computing quasipolynomials is slightly different. Assume that we want to evaluate $(\lambda', \mu') \mapsto c_{\lambda'}^{\mu'}$ in a neighborhood of a couple (λ, μ), and $(\lambda', \mu', \nu') \mapsto c_{\lambda'}{}^{\nu'}{}_{\mu'}$ in a neighborhood of a triple (λ, μ, ν). Let $\lambda_F = [x_1, \ldots, x_r]$, $\mu_F = [y_1, \ldots, y_r]$, $\nu_F = [z_1, \ldots, z_r]$, be three formal vectors where x_i's, y_i's and z_i's are variables. Then we use the command lines

```
PolynomialMultiplicityX(lambda, lambdaF, mu, muF);
PolynomialTensorProductX(lambda, lambdaF, mu, muF, nu, nuF);
```

So for the polynomial $(\lambda', \mu') \mapsto c_{\lambda'}^{\mu'}$ with $\lambda = (3, 2, 1, -6)$ and $\mu = (2, 2, -2, -2)$ for A_3 we enter

```
PolynomialMultiplicityA(
    [3, 2, 1, −6], [x[1], x[2], x[3], x[4]], [2, 2, −2, −2], [y[1], y[2], y[3], y[4]]);
```

and get instantly

$$\frac{1}{6}(3x_1 - 2y_1 + 1)(3x_1 - 2y_1 + 2)(3x_1 + 6x_2 - 2y_1 - 6y_2 + 3).$$

Remark that quasipolynomials $c_{t\lambda}^{t\mu}$ and $c_{t\lambda}{}^{t\nu}{}_{t\mu}$ are obtained by setting $x_i = t\lambda_i$, $y_i = t\mu_i$, $z_i = t\nu_i$, so that

```
PolynomialMultiplicityA(
    [3, 2, 1, −6], [3t, 2t, t, −6t], [2, 2, −2, −2], [2t, 2t, −2t, −2t]);
```

returns $(t + 1)(t + 2)(t + 3)/6$.

Now some words about implementation. There are two main parts in our programs. The first one is the implementation of Theorems 3.2 and 4.2; it is described in [**BBCV05**]. The second one is the implementation of Kostant's (2.1) and Steinberg's (2.1) formulae using valid Weyl group elements and valid couples of Weyl group elements; it is a generalization for classical Lie algebras of what has been done for A_r in [**C03**].

5.2. Comparative tests. Figure 2 describes efficiency area of the software LiE and of our programs using MNS; any area located to the left of a colored line represents the range where a program can compute examples in a reasonable time. Figures 3–5 present precise comparative tests of the software LiE, of DeLoera-McAllister's script [**DM05**] using `LattE` [**L**] and of our programs using MNS.

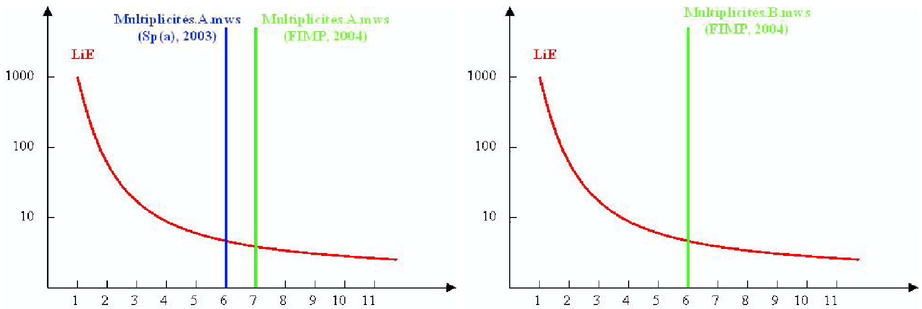

FIGURE 2. To the left, comparison for tensor product coefficients for A_r: with LiE, with Sp($\mathbf{a}$) and with MNS. To the right, comparison for weight multiplicity of a weight for B_r: with LiE and with MNS. Similar Figures for C_r and D_r.

All examples were runned on the same computer, a Pentium IV 1,13GHz with 2Go of RAM memory. Remark that computation times for `LattE` and LiE are slower than those shown in [**DM05**], due to different computers. However, we performed exactly same examples for comparison purposes.

As in [**DM05**], in Tables 3–4 weights are for $\mathfrak{gl}_{r+1}(\mathbb{C})$ and not $\mathfrak{sl}_{r+1}(\mathbb{C})$ (coordinates do not add to zero). However the sum of coordinates of $\lambda + \mu - \nu$ vanish.

Now some words about quasipolynomials computation. Let us examine the first example for B_3 in [**DM05**], that is the evaluation of the quasipolynomial ${c_{t\lambda}}^{t\nu}{}_{t\mu}$ for weights $\lambda = [0, 15, 5]$, $\mu = [12, 15, 3]$ and $\nu = [6, 15, 6]$ expressed in the basis of fundamental weights. In canonical basis, these data become $\lambda = (35/2, 35/2, 5/2)$, $\mu = (57/2, 33/2, 3/2)$, $\nu = (24, 18, 3)$. The program using the MNS algorithm returns the quasipolynomial

$$\begin{aligned}{c_{t\lambda}}^{t\nu}{}_{t\mu} \quad &= \quad \left(\frac{203}{256} + \frac{53}{256}(-1)^t\right) + \left(\frac{1515}{128} + \frac{197}{128}(-1)^t\right) t \\ &\quad + \left(\frac{35353}{384} + \frac{881}{128}(-1)^t\right) t^2 + \left(\frac{13405}{32}\right) t^3 \\ &\quad + \left(\frac{407513}{384}\right) t^4 + \left(\frac{68339}{64}\right) t^5\end{aligned}$$

in 1099,4s. On the other hand, the computation of the full quasipolynomial ${c_{\lambda}}^{\nu}{}_{\mu}$ with formal vectors $[x_1, x_2, x_3]$, $[y_1, y_2, y_3]$, $[z_1, z_2, z_3]$ leads to a 87 pages result, obtained in only 1158,6s. With `LattE`, on our computer, one obtains the quasipolynomial ${c_{t\lambda}}^{t\nu}{}_{t\mu}$ in only 825,8s.

As announced in the introduction, our program is really efficient for weights with huge coefficients. Note that in the particular case of A_r the MNS algorithm allows us to compute examples one rank further than the Sp($\mathbf{a}$) algorithm.

The translation of the program using MNS in the language of the symbolic calculation software `MuPAD` is in progress. A version using distributed calculation on a grid of computers is in the air; it will considerably increase the speed of computations.

λ, μ, ν	$c_{\lambda}{}^{\nu}{}_{\mu}$	MNS	LattE	LiE
(9,7,3,0,0), (9,9,3,2,0), (10,9,9,8,6)	2	8,0s	3,0s	$<$0,1s
(18,11,9,4,2), (20,17,9,4,0), (26,25,19,16,8)	453	2,8s	8,8s	$<$0,1s
(30,24,17,10,2), (27,23,13,8,2), (47,36,33,29,11)	5231	2,2s	11,4s	0,5s
(38,27,14,4,2), (35,26,16,11,2), (58,49,29,26,13)	16784	1,3s	12,8s	1,5s
(47,44,25,12,10), (40,34,25,15,8), (77,68,55,31,29)	5449	1,3s	8,8s	1,4s
(60,35,19,12,10), (60,54,27,25,3), (96,83,61,42,23)	13637	1,0s	8,4s	9,1s
(64,30,27,17,9), (55,48,32,12,4), (84,75,66,49,24)	49307	2,5s	9,5s	15,9s
(73,58,41,21,4), (77,61,46,27,1), (124,117,71,52,45)	557744	2,1s	12,3s	284,1s

FIGURE 3. For A_r, comparison of running times between the MNS algorithm, LattE and LiE

λ, μ, ν	$c_{\lambda}{}^{\nu}{}_{\mu}$	MNS	LattE
(935,639,283,75,48) (921,683,386,136,21) (1529,1142,743,488,225)	1303088213330	1,7s	12,8s
(6797,5843,4136,2770,707) (6071,5175,4035,1169,135) (10527,9398,8040,5803,3070)	459072901240524338	3,1s	15,1s
(859647,444276,283294,33686,24714) (482907,437967,280801,79229,26997) (1120207,699019,624861,351784,157647)	11711220003870071391294871475	2,0s	11,9s

FIGURE 4. For A_r, comparison of running times for large weights between the MNS algorithm and LattE

	λ,μ,ν	$c_{\lambda\ \mu}^{\ \nu}$	MNS	LattE	LiE
B_3	(46,42,38), (38,36,42), (41,36,44)	354440672	6,4s	22,5s	229,0s
	(46,42,41), (14,58,17), (50,54,38)	88429965	2,7s	15,2s	102,6s
	(15,60,67), (58,70,52), (57,38,63)	626863031	7,8s	17,0s	713,5s
	(5567,2146,6241), (6932,1819,8227), (3538,4733,3648)	215676881876569849679	7,0s	16,3s	–
C_3	(25,42,22), (36,38,50), (31,33,48)	87348857	5,6s	18,1s	52,9s
	(34,56,36), (44,51,49), (37,51,54)	606746767	5,1s	20,4s	516,0s
	(39,64,58), (65,15,72), (70,41,44)	519379044	8,7s	18,3s	1096,9s
	(5046,5267,7266), (7091,3228,9528), (9655,7698,2728)	1578943284716032240384	8,2s	18,3s	–
D_4	(13,20,10,14), (10,20,13,20), (5,11,15,18)	41336415	131,0s	185,8s	224,7s
	(12,22,9,30), (28,14,15,26), (10,24,10,26)	322610723	78,6s	192,7s	1184,8s
	(37,16,31,29), (40,18,35,41), (36,27,19,37)	18538329184	64,3s	258,7s	21978,4s
	(2883,8198,3874,5423), (1901,9609,889,4288), (5284,9031,2959,5527)	1891293256704574356565149344	27,7s	165,2s	–

FIGURE 5. For B_r, C_r, D_r, comparison of running times between LattE, the MNS algorithm and LiE

References

[BdLV03] Baldoni-Silva W., De Loera J.A., and Vergne M., *Counting Integer lows in Networks*, Found. Comput. Math. **4** (2004), 277–314.

[BV01] Baldoni-Silva W. and Vergne M., *Residues ormulae for Volumes and Ehrhart Polynomials of Convex Polytopes*, 81 pages, 2001, available at arXiv:math.CO/0103097.

[BBCV05] Baldoni W., Beck M., Cochet C. and Vergne M., *Volume of Polytopes and Partition unction*, 2005, 55 pages, available at arXiv:math.CO/0504231.

[BP99] Barvinok A. and Pommersheim J., *An Algorithmic Theory of Lattice Points in Polyhedra*, in: *New Perspectives in Algebraic Combinatorics* (Berkeley, CA, 1996-1997), Math. Sci. Res. Inst. Publ. **38** (1999), Cambridge Univ. Press, Cambridge, 91-147.

[B94] Barvinok A. I., *A Polynomial Time Algorithm for Counting Integral Points in Polyhedra when the Dimension is ixed*, Math. Oper. Res. **19** (1994), 769–779.

[B97] Barvinok A., *Lattice Points and Lattice Polytopes*, in *Handbook of discrete and computational geometry*, CRC Press Ser. Discrete Math. Appl., CRC, Boca Raton, FL (1997), 133–152.

[BP03] Beck M. and Pixton D., *The Ehrhart Polynomial of the Birkhoff Polytope*, Discrete Comp. Geom. **30** (2003), 623–637.

[BZ01] Berenstein A. and Zelevinsky A., *Tensor Product Multiplicities, Canonical Bases and Totally Positive Varieties*, Invent. Math. **143** (2001) 77–128.

[BGR03] Billey S., Guillemin V. and Rassart É., *A Vector Partition unction for the Multiplicities of* $\mathfrak{sl}_k(\mathbb{C})$, J. Algebra **278** (2004), 251–293.

[Bou68] N. Bourbaki, *Éléments de mathématique. asc. XXXIV. Groupes et algèbres de Lie. Chapitre IV: Groupes de Coxeter et systèmes de Tits. Chapitre V: Groupes engendrés par des réflexions. Chapitre VI: systèmes de racines.* Hermann, Paris, 1968.

[BV97] M. Brion and M. Vergne, *Residue ormulae, Vector Partition unctions and Lattice Points in Rational Polytopes*, J. Amer. Math. Soc. **10** (1997) 797–833.

[BV99] M. Brion and M. Vergne, *Arrangement of Hyperplanes. I. Rational unctions and Jeffrey-Kirwan Residue*, Ann. Sci. École Norm. Sup. (4) **32** (1999) 715–741.

[C] `http://www.math.jussieu.fr/~cochet`

[C03] Cochet C., *Kostka Numbers and Littlewood-Richardson Coefficients*, Proceedings of the AMS–IMS–SIAM Joint Summer Research Conference "Integer Points in Polyhedra – Geometry, Number Theory, Algebra, Optimization", July 13–17, 2003, Snowbird, UT, to appear in *Contempory Mathematics*.

[DCP04] De Concini C. and Procesi C., *Nested Sets and Jeffrey-Kirwan Cycles*, 10 pages, available at arXiv:math.AG/0406290.

[DHTY03] DeLoera J., Hemmecke R., Tauzer J. and Yoshida R., *Effective Lattice Point Counting in Rational Convex Polytopes*, J. Symbolic Comput. **38** (2004), no. 4, 1273–1302.

[DM05] DeLoera J. and McAllister T., *On the Computation of Clebsch–Gordan Coefficients and the Dilation Effect*, 2005, 21 pages, available at arXiv:math.RT/0501446.

[GAP] `http://www.math.rwth-aachen.de/~GAP`

[KT99] Knutson A. and Tao T., *The Honeycomb Model of* $GL_n(\mathbb{C})$ *Tensor Products. I. Proof of the saturation conjecture*, J. Amer. Math. Soc. **12** (1999), 1055–1090.

[L] `http://www.math.ucdavis.edu/~totalresidue/`

[R] `http://www.math.ias.edu.de/~rassart/presentations.html`

[Ras04] Rassart É., *A Polynomiality Property for Littlewood-Richardson Coefficients*, J. Combin. Theory Ser. A **107** (2004), 161–179.

[S95] Stembridge, J., *A Maple Package for Symmetric unctions*, J. Symbolic Computation **20** (1995), 755–768.

[SV04] Szenes A. and Vergne M., *Toric Reduction and a Conjecture of Batyrev and Materov*, Invent. Math. **158** (2004), 453–495.

[vL94] van Leeuwen M. A. A., *LiE, a Software Package for Lie Group Computations*, Euromath Bull. **1** (1994), 83–94.

Institut de Mathématiques de Jussieu, Université Paris 7 – Denis Diderot, Case 7012, 75251 Paris cedex 05, France

E-mail address: `cochet@math.jussieu.fr`