

Multiplicities and tensor product coefficients for A_r

Charles Cochet*

20th June 2003

Abstract

We apply some recent developments of Baldoni-DeLoera-Vergne [1] on vector partition functions, to Kostant and Steinberg formulas, in the case of A_r . We therefore get a fast MAPLE program that computes for A_r : the multiplicity c_λ^μ of the weight μ in the representation $V(\lambda)$ of highest weight λ ; the multiplicity $c_{\lambda,\mu}^\nu$ of the representation $V(\nu)$ in $V(\lambda) \otimes V(\mu)$. The computation also gives the locally polynomial functions c_λ^μ and $c_{\lambda,\mu}^\nu$.

1 Introduction

In this short note, we are interested in the two following problems in the case of A_r :

Mult: Computation of the multiplicity c_λ^μ of the weight μ in the representation $V(\lambda)$ of highest weight λ .

Tens: Computation of the multiplicity $c_{\lambda,\mu}^\nu$ of the representation $V(\nu)$ in the tensor product of representations of highest weights λ and μ .

The approach to these problems is through vector partition functions, namely number of integral points in lattice polytopes. More precisely, let Φ be a $n \times N$ integral matrix with column vectors Φ_1, \dots, Φ_N . Fix a n -dimensional vector a . The rational convex polytope associated to Φ and a is

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

We assume that a is in the cone $C(\Phi)$ spanned by non-negative linear combinations of the vectors Φ_i . We also assume that $\ker(\Phi) \cap \mathbb{R}_+^N = \{0\}$, so that the cone $C(\Phi)$ is acute. The *vector partition function* is then by definition

$$k(\Phi, a) = \left| P(\Phi, a) \cap \mathbb{N}^N \right|,$$

that is the number of non-negative integral solutions (x_1, \dots, x_N) of the equation $\sum_{i=1}^N x_i \Phi_i = a$. If Φ is the matrix of positive roots of A_r , then $k(\Phi, a)$ is denoted by $k(A_r^+, a)$ and it is called *Kostant partition function*. Note that Φ is the $(r+1) \times (r(r+1)/2)$ matrix with columns $e_i - e_j$ ($1 \leq i < j \leq r+1$), where e_i is the canonical basis of \mathbb{R}^{r+1} .

Let Σ_{r+1} be the set of permutations of $(r+1)$ elements. This is the Weyl group of A_r . Kostant multiplicity formula asserts that

$$c_\lambda^\mu = \sum_w (-1)^{\varepsilon(w)} k\left(A_r^+, w(\lambda + \rho) - (\mu + \rho)\right), \quad (1)$$

where ρ is half the sum of positive roots. Here, the sum is over the elements $w \in \Sigma_{r+1}$ such that $w(\lambda + \rho) - (\mu + \rho)$ is in the cone generated by non-negative combinations of positive roots. Moreover $\varepsilon(w)$ is the signature of w .

Steinberg formula asserts that

$$c_{\lambda,\mu}^\nu = \sum_{w,w'} (-1)^{\varepsilon(w)\varepsilon(w')} k\left(A_r^+, w(\lambda + \rho) + w'(\mu + \rho) - (\nu + 2\rho)\right). \quad (2)$$

Here, the sum is over couples $(w, w') \in \Sigma_{r+1} \times \Sigma_{r+1}$ such that $w(\lambda + \rho) + w'(\mu + \rho) - (\nu + 2\rho)$ is in the cone $C(A_r^+)$.

We use results of Baldoni-DeLoera-Vergne [1] and Baldoni-Vergne [2] on vector partition functions to obtain an efficient MAPLE program. Vector partition function is computed via inverse Laplace formula, involving iterated residues of rational functions.

Recall that $\mathbb{L}\mathbb{E}$ program (see [6]) uses Freudenthal and Klymik formulas. The program $\mathbb{L}\mathbb{E}$ is designed to work for any root system, while our program is designed specially for large parameters in A_r .

*cochet@math.jussieu.fr or <http://www.math.jussieu.fr/~cochet>

2 Baldoni-DeLoera-Vergne formula

Consider a $r + 1$ real dimensional vector space. Let A_r^+ (the positive root system of A_r) be defined by

$$A_r^+ = \{(e_i - e_j); 1 \leq i < j \leq (r + 1)\}.$$

Let E_r be the vector space spanned by the elements $(e_i - e_j)$. Then

$$E_r = \{a \in \mathbb{R}^{r+1}; a = a_1 e_1 + \cdots + a_r e_r + a_{r+1} e_{r+1} \text{ with } a_1 + a_2 + \cdots + a_r + a_{r+1} = 0\}.$$

The vector space E_r is of dimension r and the map

$$f : \mathbb{R}^r \longrightarrow E_r \tag{3}$$

defined by

$$a = (a_1, a_2, \dots, a_r) \longmapsto a_1 e_1 + \cdots + a_r e_r - (a_1 + \cdots + a_r) e_{r+1}$$

explicitly provides an isomorphism of E_r with the Euclidean space \mathbb{R}^r . The hyperplane arrangement (setting $z_{r+1} = 0$) generated by A_r^+ is given by the following set of hyperplanes in \mathbb{C}^r :

$$\{z_i; 1 \leq i \leq r\} \cup \{(z_i - z_j); 1 \leq i < j \leq r\}.$$

Let R_{A_r} be the set of rational functions $f(z_1, z_2, \dots, z_r)$ on \mathbb{C}^r , with poles on the hyperplanes $z_i = z_j$ or $z_i = 0$. For a permutation $w \in \Sigma_r$ define the linear form on R_{A_r}

$$\begin{aligned} \text{IRes}_{z=0}^w f &= \text{Res}_{z_{w(1)}=0} \text{Res}_{z_{w(2)}=0} \cdots \text{Res}_{z_{w(r)}=0} f(z_1, z_2, \dots, z_r) \\ &= \text{Res}_{z_1=0} \text{Res}_{z_2=0} \cdots \text{Res}_{z_r=0} f(z_{w^{-1}(1)}, z_{w^{-1}(2)}, \dots, z_{w^{-1}(r)}). \end{aligned}$$

In particular for $w = \text{id}$ the linear form $f \mapsto \text{IRes}_{z=0} f$ defined by

$$\text{IRes}_{z=0} f = \text{Res}_{z_1=0} \text{Res}_{z_2=0} \cdots \text{Res}_{z_r=0} f(z_1, z_2, \dots, z_r)$$

is called the *iterated residue*.

Let $C(A_r^+) \subset E_r$ be the cone generated by positive roots. A subset σ of A_r^+ is called a *basic* subset if $\{\sigma\}$ form a vector space basis of E_r . The *chamber complex* is the polyhedral subdivision of the cone $C(A_r^+)$ which is defined as the common refinement of the simplicial cones $C(\sigma)$ running over all possible basic subsets of A_r^+ . The pieces of this subdivision are called *chambers*. See [1] and [5] for the computation of chambers for A_r^+ .

A *wall* is a hyperplane in E_r spanned by $(r - 1)$ vectors of A_r^+ . A vector $v \in C(A_r^+)$ is *regular* if it is not on a wall. This means that for every strict subset $I \subset \{1, \dots, r + 1\}$ we have $\sum_{i \in I} v_i \neq 0$.

Let $\text{Sp}(a)$ be the set of permutations $w \in \Sigma_r$ such that:

$$\left\{ \begin{array}{lll} \text{if } a_{w(1)} \geq 0 & \text{then } w(1) < w(2) & \text{else } w(1) > w(2) \\ \text{if } a_{w(1)} + a_{w(2)} \geq 0 & \text{then } w(2) < w(3) & \text{else } w(2) > w(3) \\ \dots & & \\ \text{if } a_{w(1)} + \cdots + a_{w(i)} \geq 0 & \text{then } w(i) < w(i + 1) & \text{else } w(i) > w(i + 1) \\ \dots & & \\ \text{if } a_{w(1)} + \cdots + a_{w(r-1)} \geq 0 & \text{then } w(r - 1) < w(r) & \text{else } w(r - 1) > w(r) \end{array} \right\}$$

An element of $\text{Sp}(a)$ will be called a *special permutation* for a . Remark that if $a_i \geq 0$ for all $i \leq r$, then $\text{Sp}(a) = \{\text{id}\}$. Also remark that $\text{Sp}(a)$ is a subset of the subgroup Σ_r of the Weyl group Σ_{r+1} of A_r .

Given $a \in C(A_r^+) \cap \mathbb{Z}^{r+1}$, define $\text{def}(a) = a + \varepsilon(\sum_{i=1}^r e_i - r e_{r+1})$ with $\varepsilon = \frac{1}{2r}$.

Lemma 2.1 *The deformed vector $\text{def}(a)$ verifies:*

- $\text{def}(a)$ is regular.
- $a \in C(A_r^+)$ if and only if $\text{def}(a) \in C(A_r^+)$.

Proof:

- Let I_a be a strict subset of $\{1, \dots, r + 1\}$ such that $\sum_{i \in I_a} \text{def}(a)_i = 0$.
First, assume that $r + 1 \notin I_a$. We can re-index a in order to get $I_a = \{1, \dots, k\}$ with $k \leq r$. Thus $(a_1 + \varepsilon) + \cdots + (a_k + \varepsilon) = 0$ means that the integer $a_1 + \cdots + a_k$ equals $-\frac{k}{2r}$. But $0 < k \leq r$ implies $0 < \frac{k}{2r} < 1$, contradiction.
Now, assume that $r + 1 \in I_a$. We can also assume that $I_a = \{1, \dots, k, r + 1\}$ with $k \leq r$. By definition $(a_1 + \varepsilon) + \cdots + (a_k + \varepsilon) + (a_{r+1} - r\varepsilon)$ equals 0, therefore the integer $a_1 + \cdots + a_k + a_{r+1}$ is equal to $\frac{r-k}{2r}$. But $k \leq r$ leads to $-\frac{1}{2} < \frac{r-k}{2r} < \frac{1}{2}$, hence $k = r$. Consequently $I_a = \{1, \dots, r + 1\}$, contradiction.
- Note that the coordinates a_i of a are integers. Now the integer $a_1 + \cdots + a_i$ is non-negative if and only if $a_1 + \cdots + a_i + \frac{1}{2r}$ is non-negative, because $0 < \frac{1}{2r} < 1$. Hence $a \in C(A_r^+)$ is equivalent to $\text{def}(a) \in C(A_r^+)$.

Now we can state the formula that was implemented:

Theorem 2.2 (Baldoni-DeLoera-Vergne [1]) For $a \in C(A_r^+) \cap \mathbb{Z}^{r+1}$, the Kostant partition function is given by:

$$k(A_r^+, a) = \sum_{w \in Sp(a')} (-1)^{n(w)} \text{IRes}_{z=0}^w \left(\frac{(1+z_1)^{a_1+r-1} (1+z_2)^{a_2+r-2} \cdots (1+z_r)^{a_r}}{z_1 \cdots z_r \prod_{1 \leq i < j \leq r} (z_i - z_j)} \right)$$

where

$$a' = \begin{cases} a & \text{if } a \text{ is regular,} \\ \text{def}(a) & \text{otherwise.} \end{cases}$$

In particular, if $a_i \geq 0$ for $1 \leq i \leq r$, we have

$$k(A_r^+, a) = \text{Res}_{z_1=0} \text{Res}_{z_2=0} \cdots \text{Res}_{z_r=0} \left(\frac{(1+z_1)^{a_1+r-1} (1+z_2)^{a_2+r-2} \cdots (1+z_r)^{a_r}}{z_1 \cdots z_r \prod_{1 \leq i < j \leq r} (z_i - z_j)} \right).$$

3 Deus ex machina

This section features a brief description of the algorithms that were implemented with the software MAPLE. This program is available at <http://www.math.jussieu.fr/~cochet>

3.1 How to use the program

The initial data are only vectors: two for computing the multiplicity c_λ^μ , three for computing the tensor product coefficient $c_{\lambda,\mu}^\nu$.

Our program works with weights represented in the canonical basis of \mathbb{R}^{r+1} , and not fundamental weights basis of A_r like `LE`. The translation between these two approaches is performed via the procedures `fundamental` and `fundamental_inverse`. For example `fundamental([2,1,-3])` returns [1,4].

Therefore computing the multiplicity c_λ^μ is done by typing in `multiplicity(lambda,mu)` where λ and μ are lists of $r+1$ rationals such that $\sum_{i=1}^{r+1} \lambda_i = \sum_{i=1}^{r+1} \mu_i$ and $\lambda_i - \lambda_{i+1} \in \mathbb{N}$, $\mu_i - \mu_{i+1} \in \mathbb{Z}$.

For computing the tensor product coefficient $c_{\lambda,\mu}^\nu$, the syntax is `tensor_product(lambda,mu,nu)` where λ , μ and ν are lists of $r+1$ rationals such that $\sum_{i=1}^{r+1} (\lambda_i + \mu_i) = \sum_{i=1}^{r+1} \nu_i$ and $\lambda_i - \lambda_{i+1} \in \mathbb{N}$, $\mu_i - \mu_{i+1} \in \mathbb{N}$, $\nu_i - \nu_{i+1} \in \mathbb{N}$.

In the examples, we use the vector $\theta = r e_1 + (r-1)e_2 + \cdots + 1e_{r-1} - \frac{r(r+1)}{2}e_{r+1}$. Its decomposition in the fundamental weights basis is the r -dimensional vector $(1, \dots, 1, 1 + r(r+1)/2)$.

3.2 Implementation

The elements we need to compute are:

1. The vector $a' = \text{def}(a)$ obtained by deforming the initial parameter a .
2. The residues that appear in theorem 2.2.
3. The two sets of permutations that appear in Kostant and Steinberg formulas (see (1) and (2)).
4. The set of special permutations $\text{Sp}(\text{def}(a))$.

Because of lemma 2.1, we may use `def(a)` instead of a and we do this to simplify the procedures. We compute the vector `def(a)` via the straightforward MAPLE procedure `defvector`. This takes care of the first part.

Computation of residues is done iteratively. The function F which residue we need to compute is a product of a certain number of functions. This allows to take the residues by introducing little by little the part of the function F containing the needed variable. See a detailed explanation of this procedure in [1].

Let $u, v \in E^r$. A *valid permutation* for u and v is a permutation $w \in \Sigma_{r+1}$ such that $w(u) - v \in C(A_r^+)$. We denote by $V(u, v)$ the set of valid permutations for u, v . Hence, Kostant formula for A_r rewrites as

$$c_\lambda^\mu = \sum_{w \in V(\lambda+\rho, \mu+\rho)} (-1)^{\varepsilon(w)} k(A_r^+, w(\lambda+\rho) - (\mu+\rho)).$$

Given a set of chambers $\{C_w\}_{w \in V(\lambda, \mu)}$ of $C(A_r^+)$, it follows from [7] that c_λ^μ is polynomial when $w\lambda - \mu \in \frac{C_w}{2}$, for $w \in V(\lambda, \mu)$. In particular, the function $N \mapsto c_{N\lambda, N\mu}$ is a polynomial in N of degree less or equal to $\frac{r(r-1)}{2}$.

Let us explain our implementation with the symbolic language MAPLE of the procedure `valid_permutations` designed to find the set $V(u, v)$. The method is quite simple: we build the permutations iteratively. This allows us not examining all permutations and saving much time. Recall that we have to find all permutations w 's such that $u_{w(1)} \geq v_1$, $u_{w(1)} + u_{w(2)} \geq v_1 + v_2$, etc. For any sequence x of indices, we denote by u_x the sum $\sum_{i \in x} u_i$.

Step 1. Let X be the set of all indices i such that $u_i \geq v_1$.

Step 2. For each $x \in X$, we find all indices i_x such that $u_x + u_{i_x} \geq v_x + v_{i_x}$. Let X_{new} be the set of such $[x, i_x]$, for all x and i_x . Then $X \leftarrow X_{new}$.

We repeat r times step 2, and obtain the list X of $(r+1)$ -uples representing permutations of $1, \dots, r+1$. The second step is treated in the procedure `next_index_valid_permutations`. The procedure `valid_permutations` contains first step and a `for...do` loop executing r times step 2.

Remark 3.1 *We reduce computing time by using the following three tricks.*

1. We compute once and for all the vector $v' = [v_1, v_1 + v_2, \dots, v_1 + \dots + v_{r+1}]$.
2. We build at the same time of $X = [[i_1, \dots, i_p], \dots, (\text{other sets of indices})]$ the set SX of partial sums associated to each $[i_1, \dots, i_p]$. More precisely $SX = [u_{i_1} + \dots + u_{i_p}, \dots, (\text{other partial sums})]$.
3. We use tables instead of lists.

Now let us examine the couples of permutations involved in Steinberg formula. Let $u_1, u_2, v \in E^r$. A valid couple of permutations for u_1, u_2 and v is a couple $(w_1, w_2) \in \Sigma_{r+1} \times \Sigma_{r+1}$ such that $w_1(u_1) + w_2(u_2) - v \in C(A_r^+)$. We denote by $V(u_1, u_2, v)$ the set of valid couples of permutations for u_1, u_2 and v . Hence Steinberg formula rewrites as

$$c_{\lambda, \mu}^\nu = \sum_{(w, w') \in V(\lambda + \rho, \mu + \rho, \nu + 2\rho)} (-1)^{\varepsilon(w)\varepsilon(w')} k(A_r^+, (w(\lambda + \rho) + w'(\mu + \rho) - (\nu + 2\rho))).$$

The procedure computing valid couples of permutations is similar to the former.

To compute the subset $\text{Sp}(a)$ of Σ_r , we use the procedure `special_permutations`. This procedure is very similar to the previous one. We stress that the MAPLE function `combinat[permute]` is impractical and does not go very far because of memory limitations.

4 Test of the program

Let θ be the r -dimensional vector $(1, \dots, 1, 1 + r(r+1)/2)$ (fundamental weights decomposition in A_r). It translates as $(r, r-1, \dots, 1, -r(r+1)/2)$ in the canonical basis of \mathbb{R}^{r+1} . We used this vector to check the well-known fact that the multiplicity of the weight 0 in the representation of A_r of highest weight $N\theta$ is given by the dimension of the representation of A_{r-1} of highest weight $N\rho$, which is $(N+1)^{r(r-1)/2}$.

In this test, we compute for various A_r ($r = 1, \dots, 8$):

- c_λ^μ with $\lambda = N\theta$ and either $\mu = 0$ (worst case), or $\mu = [9N/10]\theta$ (intermediate case).
- $c_{\lambda, \mu}^\nu$ with $\lambda = \mu = N\theta$ and either $\nu = 0$ (worst case), or $\nu = 2[9N/10]\theta$ (intermediate case).

Tests were made with bi-processor PIII 1, 13GHz. The notation "–" in an array means that we did not try the computation (and not that computation failed).

Recall (see for example [4] and [3]) that counting integral points in a lattice polytope is polynomial in the size of input if dimension is fixed, and NP-hard if dimension is not fixed. The figures 1 and 2 emphasizes this result.

In figure 1, the letter I stands for intermediate case ($\mu = [9N/10]\theta$ for c_λ^μ and $\nu = 2[9N/10]\theta$ for $c_{\lambda, \mu}^\nu$), while W stands for worst case ($\mu = 0$ for c_λ^μ and $\nu = 0$ for $c_{\lambda, \mu}^\nu$).

| | $N = 10^1$ | $N = 10^2$ | $N = 10^3$ | $N = 10^4$ | $N = 10^5$ | $N = 10^6$ | $N = 10^7$ | $N = 10^8$ | $N = 10^9$ |
|-----------------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| c_λ^μ , I, A_7 | 12.5 s | 16.0 s | 17.5 s | 18.7 s | 17.63 s | 19.4 s | 20.3 s | 21.3 s | 22.5 |
| c_λ^μ , W, A_7 | 204.9 s | 221.0 s | 235.6 s | 251.5 s | 259.1 s | 261.5 s | 283.8 s | 297.0 s | 297.2 s |
| $c_{\lambda, \mu}^\nu$, I, A_6 | 40.5 s | 47.0 s | 50.3 s | 52.6 s | 53.8 s | 57.0 s | 58.4 s | 59.9 s | 62.3 s |
| $c_{\lambda, \mu}^\nu$, W, A_4 | 13.5 s | 13.7 s | 13.8 s | 14.0 s | 14.1 s | 14.4 s | 15.1 s | 15.2 s | 15.5 s |

Figure 1: Time of computation, when size of input grows

The computation can also be done with parameters, giving $(N + 1)^{r(r-1)/2}$ as expected.

| Algebra | Time | Multiplicity c_θ^0 | Time | Polynomial $N \mapsto c_{N\theta}^0$ |
|---------|---------|---------------------------|---------|--------------------------------------|
| A_2 | < 0.1 s | $2 = 2^1$ | < 0.1 s | $(N + 1)^1$ |
| A_3 | < 0.1 s | $8 = 2^3$ | < 0.1 s | $(N + 1)^3$ |
| A_4 | < 0.1 s | $64 = 2^6$ | < 0.1 s | $(N + 1)^6$ |
| A_5 | 0.4 s | $1024 = 2^{10}$ | 1.4 s | $(N + 1)^{10}$ |
| A_6 | 7.6 s | $32768 = 2^{15}$ | 36.2 s | $(N + 1)^{15}$ |
| A_7 | 169.3 s | $2097152 = 2^{21}$ | 2091 s | $(N + 1)^{21}$ |
| A_8 | 9401 s | $268435456 = 2^{28}$ | – | – |

Figure 2: Multiplicity of 0 in $V(N\theta)$ when rank increases

Acknowledgements: Marc A. A. van Leeuwen explained me his software `LiE` and its internal mechanisms. Moreover, his competence in `MAPLE` has deeply influenced my method of programming.

References

- [1] **Baldoni-Silva W., De Loera J.A., and Vergne M.**, *Counting Integer flows in Networks*, available at math.ArXiv, CO/0303228. Program available at <http://www.math.ucdavis.edu/~totalresidue>
- [2] **Baldoni-Silva W. and Vergne M.**, *Residues formulae for volumes and Ehrhart polynomials of convex polytopes*. manuscript 81 pages, 2001. available at math.ArXiv, CO/0103097.
- [3] **Barvinok A. and Pommersheim J.**, *An algorithmic theory of lattice points in polyhedra*, in: *New Perspectives in Algebraic Combinatorics* (Berkeley, CA, 1996-1997), 91-147, Math. Sci. Res. Inst. Publ. 38, Cambridge Univ. Press, Cambridge, 1999.
- [4] **Barvinok A.**, *Lattice points and lattice polytopes*, in *Handbook of discrete and computational geometry*, CRC Press Ser. Discrete Math. Appl., pages 133–152. CRC, Boca Raton, FL, 1997.
- [5] **De Loera J.A. and Sturmfels M.**, *Algebraic Unimodular Counting*, math.CO/0104286
- [6] **Van Leeuwen M. A. A.**, *LiE, a software package for Lie group computations*, *Euromath Bull.* 1 (1994), no. 2, p. 83–94.
- [7] **Szenes A. and Vergne M.**, *Residue formulae for vector partitions and Euler-MacLaurin sums*. preprint (2002), 52 pages. Available at math.ArXiv, CO/0202253.