On the Computation of Clebsch–Gordan Coefficients and the Dilation effect *

Jesús A. De Loera and Tyrrell B. McAllister

January 19, 2005

Abstract

We investigate the problem of computing tensor product multiplicities for complex semisimple Lie algebras. Even though computing these numbers is $\#P$-hard in general, we show that if the rank of the Lie algebra is assumed fixed, then there is a polynomial time algorithm, based on counting the lattice points in polytopes. In fact, for Lie algebras of type A_r, there is an algorithm, based on the ellipsoid algorithm, to decide when the coefficients are nonzero in polynomial time for arbitrary rank. Our experiments show that the lattice point algorithm is superior in practice to the standard techniques for computing multiplicities when the weights have large entries but small rank. Using an implementation of this algorithm, we provide experimental evidence for conjectured generalizations of the saturation property of Littlewood–Richardson coefficients. One of these conjectures seems to be valid for types B_n, C_n, and D_n.

1 Introduction

Given highest weights λ, μ, and ν for a finite dimensional complex semisimple Lie algebra, we denote by $C^{\nu}_{\lambda\mu}$ the multiplicity of the irreducible representation V_ν in the tensor product of V_λ and V_μ; that is, we write

$$V_\lambda \otimes V_\mu = \bigoplus_{\nu} C^{\nu}_{\lambda\mu} V_\nu. \tag{1}$$

In general, the numbers $C^{\nu}_{\lambda\mu}$ are called *Clebsch–Gordan coefficients*. In the specific case of type A_r Lie algebras, the values $C^{\nu}_{\lambda\mu}$ defined in equation (1) are called *Littlewood–Richardson coefficients*. When we are specifically discussing the type A_r case, we will adhere to convention and write $c^{\nu}_{\lambda\mu}$ for $C^{\nu}_{\lambda\mu}$.

*Research supported by NSF Grants DMS-0309694 and VIGRE Grant DMS-0135345. The first author is also grateful for support received from the Alexander von Humboldt foundation and a UC Davis Chancellor's fellow award.

The concrete computation of Clebsch–Gordan coefficients (sometimes known as the *Clebsch–Gordan problem* [21]) has attracted a lot of attention from not only representation theorists, but also from physicists, who employ them in the study of quantum mechanics (e.g. [10, 37]). The importance of these coefficients is also evidenced by their widespread appearance in other fields of mathematics besides representation theory. For example, the Littlewood–Richardson coefficients appear in combinatorics via symmetric functions and in enumerative algebraic geometry via Schubert varieties and Grassmannians (see for instance [33, 22]). More recently, Clebsch–Gordan coefficients are playing an important role on the study of P vs. NP (see [26]). Very recently, Narayanan has proved that the computations of Clebsch–Gordan coefficients is in general $\#P$-complete [27]. Here are our contributions:

(1) We combine the lattice point enumeration algorithm of Barvinok [3] with polyhedral results due to Knutson and Tao [19] and Berenstein and Zelevinsky [6] in the polyhedral realization of Clebsch–Gordan coefficients to produce a new algorithm for computing these coefficients. Our main theoretical result is:

Theorem 1.1. *For fixed rank r, if $\mathfrak{g}$ is a complex semisimple Lie algebra of rank r, then one can compute Clebsch–Gordan coefficients for $\mathfrak{g}$ in time polynomial in the input size of the defining weights.*

Moreover, as a consequence of the polynomiality of linear programming and the saturation property of Lie Algebras of type A_r, deciding whether $c_{\lambda\mu}^{\nu} = 0$ can be done in polynomial time, even when the rank is not fixed.

(2) We implemented the algorithm for types A_r, B_r, C_r, and D_r (the so-called "classical" Lie Algebras) using the software packages `LattE` and `Maple`. In many instances, our implementation performs faster than standard methods, such as those implemented in the software `LiE`. Our software is freely available at `http://math.ucdavis.edu/~tmcal`.

(3) Via computer experiments, we explored general properties satisfied by the Clebsch–Gordan coefficients for the classical Lie algebras under the operation of *stretching of multiplicities* in the sense of [18]. On the basis of abundant experimental evidence, we propose two conjectured generalizations of the Saturation Theorem of Knutson and Tao [19]. One of them, which applies to all of the classical root systems, is an extension of earlier work by King et al. ([18]).

Organization of the paper: In Section 2, after a review of the background material, we prove Theorem 1.1. Section 3 explains our experiments comparing our software, a mixture of `Maple` and `LattE` [13], with `LiE`. In Section 4, we present the two conjectures, both of which, if true, would generalize the Saturation Theorem of Knutson and Tao.

2 Clebsch–Gordan coefficients: Polyhedral Algorithms

As stated in the introduction, we are interested in the problem of efficiently computing $C_{\lambda\mu}^{\nu}$ in the tensor product expansion $V_\lambda \otimes V_\mu = \bigoplus_\nu C_{\lambda\mu}^{\nu} V_\nu$. It appears that the most common method used to compute the Clebsch–Gordan coefficients is based on Klimyk's formula (see lemma below). For example, it is used in `LiE` [36] and the `Maple` [25] packages `Coxeter` and `Weyl` [34].

Lemma 2.1. *([16], Exercise. 24.9) Fix a complex semisimple Lie algebra $\mathfrak{g}$, and let $\mathfrak{W}$ be the associated Weyl group. For each weight ν of $\mathfrak{g}$, let $\operatorname{sgn}(\nu)$ denote the parity of the minimum length of an element $\sigma \in \mathfrak{W}$ such that $\sigma(\nu)$ is a highest weight, and let $\{\nu\}$ denote that highest weight. Let δ be one-half the sum of the positive simple roots of $\mathfrak{g}$. Finally, for each highest weight λ of $\mathfrak{g}$, let $K_{\lambda\nu}$ be the multiplicity of ν in V_λ.*

Then, given highest weights λ and μ of $\mathfrak{g}$, we have that

$$V_\lambda \otimes V_\mu = \bigoplus_{\varepsilon} K_{\lambda\varepsilon} \operatorname{sgn}(\varepsilon + \mu + \delta) V_{\{\varepsilon+\mu+\delta\}-\delta},$$

where the sum is over weights ε of $\mathfrak{g}$ with trivial stabilizer subgroup in $\mathfrak{W}$.

Implementations of Klimyk's algorithm begin by computing the weight spaces appearing with nonzero multiplicity in the representation V_λ. Then, for each such weight ε with trivial stabilizer, one computes the Weyl group orbit of $\varepsilon + \mu + \delta$. One then finds the dominant member of the orbit and notes the number l of reflections needed to reach it. Finally, one adds $(-1)^l K_{\lambda\varepsilon}$ to the multiplicity of $V_{\{\varepsilon+\mu+\delta\}-\delta}$.

From the point of view of computational complexity, Klimyk's algorithm has two main disadvantages. First, it requires the computation of weight space multiplicities, which is in general a $\#P$-hard problem [27]. The second disadvantage is that the algorithm requires visiting all of the orbit members, which can be an exponentially large set. Indeed, Klimyk's formula above is exponential in the size of the input weights. Thus, in practice, the sizes of λ, μ, and ν usually need to be small. One can then ask for an algorithm that behaves well as the sizes of the input weights increase, at least if some other parameter is fixed. Stembridge also raised the challenge of crafting algorithms based on geometric ideas such as Littelmann's paths or Kashiwara's crystal bases [23, 24] (see comment on page 29, section 7, of [35]). As we see below, there is such an algorithm, based on the polyhedral geometry of the Clebsch–Gordan coefficients.

In 1992, Berenstein and Zelevinsky presented a combinatorial interpretation of the Littlewood–Richardson coefficients as the number of lattice points in members of a certain family of polytopes [5]. In 1998, Knutson and Tao introduced another family, the *hive polytopes*, which they used to prove the Saturation Theorem (see [19] and Section 4). Each of the polytopes presented by Berenstein and Zelevinsky in 1992 is the image under an injective lattice-preserving linear map of a hive polytope [28]. Therefore $c_{\lambda\mu}^{\nu}$ equals the number of integer lattice points in a corresponding hive polytope $H_{\lambda\mu}^{\nu}$. Finally, in 2001, Berenstein and Zelevinsky [6] introduced polytopes which enumerate Clebsch–Gordan coefficients for any finite dimensional complex semisimple Lie algebra. We refer to this last family of polytopes as the *BZ–polytopes*. We now give the exact definition of the hive polytopes. These polytopes exist in the polyhedral cone of *hive patterns*, which we now define.

Definition 2.2. Fix $r \in \mathbb{Z}_{\geq 0}$ and let $\mathcal{H} = \{(i,j,k) \in \mathbb{Z}_{\geq 0}^3 : i+j+k = r\}$. A *hive pattern* is a map

$$h\colon \mathcal{H} \to \mathbb{R}_{\geq 0}, \qquad (i,j,k) \mapsto h_{ijk},$$

satisfying the *rhombus inequalities*:

- $h_{i,j-1,k+1} + h_{i-1,j+1,k} \leq h_{ijk} + h_{i-1,j,k+1}$,
- $h_{ijk} + h_{i-1,j-1,k+2} \leq h_{i,j-1,k+1} + h_{i-1,j,k+1}$,
- $h_{i+1,j-1,k} + h_{i-1,j,k+1} \leq h_{ijk} + h_{i,j-1,k+1}$.

for $(i,j,k) \in \mathcal{H}$, $i,j \geq 1$.

Equivalently, a hive pattern of rank r is a triangular array of real numbers

$$\begin{array}{ccccccccc}
 & & & & h_{0,0,r} & & & & \\
 & & & h_{1,0,r-1} & & h_{0,1,r-1} & & & \\
 & & h_{2,0,r-2} & & h_{1,1,r-2} & & h_{0,2,r-2} & & \\
 & \iddots & & & \vdots & & & \ddots & \\
h_{r,0,0} & & h_{r-1,1,0} & & \cdots & & h_{1,r-1,0} & & h_{0,r,0}
\end{array}$$

such that, in every "little rhombus" of entries

$$\begin{array}{ccc}
c & & b \\
 & & \\
 & a & & d
\end{array}, \qquad
\begin{array}{ccc}
 & c & \\
a & & b \\
 & d &
\end{array}, \qquad
\begin{array}{cccc}
 & a & & c \\
 & & & \\
d & & b &
\end{array}$$

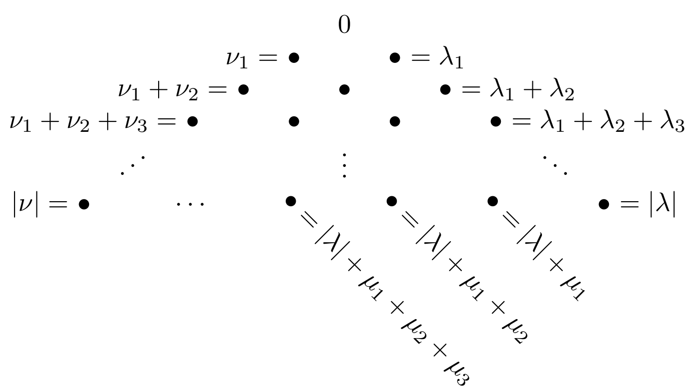
we have $a + b \geq c + d$.

Here is example of a hive pattern with $r = 5$:

$$
\begin{array}{ccccccccc}
 & & & & 0 & & & & \\
 & & & 5 & & 8 & & & \\
 & & 8 & & 12 & & 13 & & \\
 & 11 & & 15 & & 17 & & 18 & \\
12 & & 16 & & 18 & & 20 & & 20
\end{array}
$$

Recall that when $\mathfrak{g}$ is of type A_r, so that $\mathfrak{g} \cong \mathfrak{sl}_{r+1}(\mathbb{C})$ for some $r \geq 2$, the highest weights are (with respect to the canonical basis) partitions of length r, *i.e.*, sequences λ of integers $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. We write $|\lambda|$ for $\sum_i \lambda_i$, the *size* of the partition λ.

Definition 2.3. Given partitions $\lambda, \mu, \nu \in \mathbb{Z}^n_{\geq 0}$, the *hive polytope* $H^\nu_{\lambda\mu}$ is the set of hive patterns with boundary entries fixed as below.



Note that, for fixed r, the input size of a hive polytope $H^\nu_{\lambda\mu}$ grows linearly with the input sizes of the weights λ, μ, and ν. We will need the following result:

Lemma 2.4. *([20]) The Littlewood–Richardson coefficient $c^\nu_{\lambda\mu}$ equals the number of integer lattice points in the hive polytope $H^\nu_{\lambda\mu}$.*

Unfortunately, the description of the BZ-polytopes is more involved than that of the hive polytopes above. Therefore, we refer the reader to Theorems 2.5 and 2.6 of [6], which give their description as systems of linear equalities and inequalities in terms of the root systems B_r, C_r, and D_r. The reader may also view the contents of our maple notebook, available at `http://math.ucdavis.edu/~tmcal`, for completely explicit descriptions of the necessary inequalities. The specific properties of the BZ-polytopes that we need to prove our theorem are (1) for fixed rank r, the dimensions of the BZ-polytopes are bounded above by a constant, (2) the input size of a BZ-polytope grows linearly with the input sizes of the weights λ, μ, and ν, and (3) the following result describing the relationship between BZ-polytopes and Clebsch–Gordan coefficients:

Lemma 2.5. *(Theorem 2.4 of [6]) Fix a finite dimensional complex semi-simple Lie algebra $\mathfrak{g}$ and a triple of highest weights (λ, μ, ν) for $\mathfrak{g}$. Then the Clebsch–Gordan coefficient $C_{\lambda\mu}^{\nu}$ equals the number of integer lattice points in the corresponding BZ-polytope.*

Finally, the last ingredient necessary is A. Barvinok's algorithm for counting lattice points in polytopes in polynomial time for fixed dimension. Several detailed descriptions of the algorithm in Lemma 2.6 are now available the literature (see [12] and all the references therein).

Lemma 2.6. *([3]) Fix $d \in \mathbb{Z}_{\geq 0}$. Then, given a system of equalities and inequalities defining a rational convex polytope $P \subset \mathbb{R}^d$, we can compute $\#(P \cap Z^d)$ in time polynomial in the input size of the polytope.*

Proof of Theorem 1.1: First, if we fix the rank of the Lie algebra, then we fix an upper bound on the dimension of the hive or BZ polytope. Moreover, the input sizes of these polytopes grow linearly with the input sizes of the weights. Thus, by Barvinok's theorem (Lemma 2.6 stated above), their lattice points can be computed in time polynomial in the input sizes of the weights. Therefore, the theorem follows by Lemmas 2.4 and 2.5.

For the second part of the theorem regarding type A_r, the hive polytopes provide a very fast method for determining whether $c_{\lambda\mu}^{\nu} = 0$. According to the Saturation Theorem (see Section 4), $c_{\lambda\mu}^{\nu} = 0$ if and only if the corresponding hive polytope is empty. Hence, it suffices to check whether the system of inequalities defining the hive polytope is feasible, which can be done in polynomial time for *arbitrary* dimension as a corollary of the polynomiality of linear programming via Khachian's ellipsoid algorithm (see [30]).

It is useful to notice that every Kostka number $K_{\lambda\mu}$ is a Littlewood–Richardson coefficient for some choice of highest weights. One short bijection is given by $K_{\lambda\mu} = c_{\sigma\lambda}^{\tau}$, where

$$\begin{cases} \tau_i = \mu_i + \mu_{i+1} + \cdots \\ \sigma_i = \mu_{i+1} + \mu_{i+2} + \cdots \end{cases} \qquad \text{for } i = 1, 2, \ldots, r.$$

For those familiar with the enumeration of semi-standard Young tableaux and Littlewood–Richardson tableaux by Kostka numbers and Littlewood–Richardson coefficients respectively (see, *e.g.*, [32]), the bijection is straightforward: Given a semi-standard Young tableau Y with shape λ and content μ, construct a Littlewood–Richardson tableau L with shape τ/σ and content λ by filling the boxes as follows. Start with a skew Young diagram D with shape τ/σ. For $j = 1, \ldots, r$, and for each entry i in the jth row of Y, place a j in the ith row of D. Let L be the tableau produced by filling the boxes of D in this fashion. (See Figure 1.) It is not hard to see that, under this map, the column-strictness condition on Y is equivalent to the lattice permutation condition on L. It follows that the map just described is a bijection between semi-standard Young tableaux

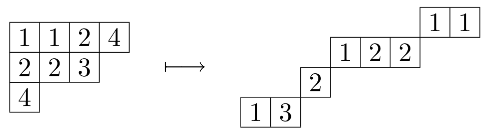


Figure 1: Corresponding semi-standard Young and Littlewood–Richardson tableaux

with shape λ and content μ and Littlewood–Richardson tableaux with shape τ/σ and content λ. Thus, computing Kostka numbers reduces to computing Littlewood–Richardson coefficients.

As a corollary, we get a similar result for the computation of Kostka numbers based on polyhedral methods.

Corollary 2.7. *For fixed rank r, the Kostka number $K_{\lambda\mu}$ can be computed in polynomial time in the size of the input weights λ and μ. For arbitrary rank one can decide in polynomial time in the size of the weights whether $K_{\lambda\mu} = 0$ or not.*

3 Using the Algorithm in Practice

Using the explicit definitions for the hive and BZ-polytopes as the sets of solutions to systems of linear inequalities and equalities, we wrote a `Maple` notebook which, when given a triple of highest weights, produces the corresponding hive or BZ–polytope in `LattE` readable input format. The notebook is available from `http://math.ucdavis.edu/~tmcal`. All computations were done on a Linux PC with a 2 GHZ CPU and 4 Gigabytes of memory.

From our experiments, we conclude that (1) The polyhedral method of computing tensor product multiplicities complements the method employed in `LiE`. `LiE` is effective for larger ranks (up to $r = 10$, say), but the sizes of the weights must be kept small. This is because `LiE` uses the Klimyk formula to generate the entire direct sum decomposition of the tensor product, after which it dispenses with all but the single desired term. However, computing all of the terms in the direct sum decomposition is not feasible when the sizes of the entries in the weights grow into the 100s. On the other hand, (2) Lattice point enumeration is often effective for very large weights (in particular, the algorithm is suitable for investigating the stretching properties of Section 4). However, the rank must be relatively low (roughly $r \leq 6$) because lattice point enumeration complexity grows exponentially in the dimension of the polytope, and the dimensions of these polytopes grow quadratically with the rank of the Lie algebra. Together, the two algorithms cover a larger range of problems.

We would also like to mention that other authors are also using lattice points in polytopes to compute Clebsch–Gordan coefficients, this time via the calculation of vector partition functions for classical root systems [2, 8, 9]. These authors report that, like us, they can compute with large sizes of weight entries.

3.1 Experiments for type A_r

In the tables below, we express highest weights in terms of the canonical basis $e_1, \ldots, e_r$, so that the highest weights are partitions with r parts.

Experiments indicate that lattice point enumeration is very efficient for computing Littlewood–Richardson numbers when $r \leq 5$. First, we computed over 30 instances with randomly generated weights with leading entries larger than 40 with our approach and with `LiE`. In all cases our algorithm was faster. After that, we did a "worst case" sampling for the table in Figure 2 comparing the computation times of `LattE` and `LiE`. To produce the ith row of that table, we selected uniformly at random 1000 triples of weights (λ, μ, ν) in which the largest parts of λ and μ were bounded above by $10i$ and $|\nu| = |\lambda| + |\mu|$ (this is a necessary condition for $c^{\nu}_{\lambda,\mu} \neq 0$). Then we evaluated the corresponding hive polytopes with `LattE`. The `LattE` input files are created with our `Maple` program. The weight triple in the ith row is the one that `LattE` took the longest time to compute. We then computed the same tensor product multiplicity with `LiE`. The table in Figure 3, shows the running time needed when using `LattE` to compute weight triples with entries in the thousands or millions.

When $r \geq 6$, the running time under `LattE` begins to blow up. Still, for $r = 6$, all examples we attempted could be computed in under 30 minutes using `LattE`, and most could be computed in under 5 minutes. For example, among 54 nonempty hive polytopes chosen uniformly at random among those in which the weights had entries less than 100, all but seven could be computed in under 5 minutes with `LattE`, and the remaining seven could all be computed in under 30 minutes. None of these Littlewood–Richardson coefficients could be computed with `LiE`. At $r = 7$, lattice point enumeration becomes less effective, with examples typically taking several hours or more to evaluate.

3.2 Experiments for types B_r, C_r, and D_r

To compute Clebsch–Gordan coefficients in types B_r, C_r, and D_r, we used the BZ-polytopes. In the tables that follow, all weights are given in the basis of fundamental weights for the corresponding Lie algebra.

Our experiments followed the same process we used for A_r: First, for each root system, we computed over 30 instances with randomly generated weights with entries larger than 40 with our approach and with `LiE`. In all cases our algorithm was faster. After that, we did a "worst case" sampling to produce the tables in Figure 4 comparing the computation times of `LattE` and `LiE`. As in Section 3.1, these weight triples were the ones which `LattE` took the longest to evaluate among thousands of instances generated with the following procedure:

λ, μ, ν	$c_{\lambda\mu}^{\nu}$	`LattE` runtime	`LiE` runtime
(9,7,3,0,0), (9,9,3,2,0), (10,9,9,8,6)	2	0m00.74s	0m00.01s
(18,11,9,4,2), (20,17,9,4,0), (26,25,19,16,8)	453	0m03.86s	0m00.12s
(30,24,17,10,2), (27,23,13,8,2), (47,36,33,29,11)	5231	0m05.21s	0m02.71s
(38,27,14,4,2), (35,26,16,11,2), (58,49,29,26,13)	16784	0m06.33s	0m25.31s
(47,44,25,12,10),(40,34,25,15,8),(77,68,55,31,29)	5449	0m04.35s	1m55.83s
(60,35,19,12,10),(60,54,27,25,3),(96,83,61,42,23)	13637	0m04.32s	23m32.10s
(64,30,27,17,9), (55,48,32,12,4), (84,75,66,49,24)	49307	0m04.63s	45m52.61s
(73,58,41,21,4), (77,61,46,27,1), (124,117,71,52,45)	557744	0m07.02s	$>$ 24 hours

Figure 2: A sample comparison of running times between `LattE` and `LiE` case of A_r

λ, μ, ν	$c_{\lambda\mu}^{\nu}$	`LattE` runtime
(935,639,283,75,48) (921,683,386,136,21) (1529,1142,743,488,225)	1303088213330	0m07.84s
(6797,5843,4136,2770,707) (6071,5175,4035,1169,135) (10527,9398,8040,5803,3070)	459072901240524338	0m09.63s
(859647,444276,283294,33686,24714) (482907,437967,280801,79229,26997) (1120207,699019,624861,351784,157647)	11711220003870071391294871475	0m08.15s

Figure 3: Computing large weights with `LattE`. Case of A_r

First, to produce line i of a table, we set an upper bound U_i for the entries of each weight. Then, we generated 1000 random weight triples with entry sizes no larger than U_i. Here are the specific values of U_i used in each of the three tables in Figure 4. For type B_r, the bounds U_i were 50, 60, 70, and 10,000, respectively. For type C_r, the bounds U_i were 50, 60, 80, and 10,000, respectively. Finally, for type D_r, the bounds U_i were 20, 30, 40, and 10,000, respectively. For each generated triple of weights, we produced the associated BZ-polytopes (using our `Maple` notebook) and counted their lattice points with `LattE`. In the table are those instances that were slowest in `LattE`. We also recorded in the table the time taken by `LiE` for the same instances. One can see the running time needed by `LattE` is hardly affected by growth in the size of the input weights, while the time needed by `LiE` grows rapidly.

We found that for types B_r and C_r, lattice point enumeration with the BZ-polytopes is very effective when $r \leq 3$. Each of the many thousands of examples we generated could be evaluated by `LattE` in under 10 seconds (the examples in Figure 4 were the worst cases). When $r = 4$, the computation time begins to blow up, with examples typically taking half an hour or more to compute. The polyhedral method is also reasonably efficient for type D Lie algebras with rank 4, the lowest rank in which they are defined. All of the examples we generated could be evaluated by `LattE` in under 5 minutes.

4 Two Conjectures that Could Generalize the Saturation Theorem

In 1998, Knutson and Tao used the hive polytopes to prove the *Saturation Theorem*. Buch has written a very clear exposition of this proof in [7].

Theorem 4.1 ([19]). *(Saturation) Given highest weights λ, μ, and ν for a Lie algebra of type A_r, and given an integer $n > 0$, the Littlewood–Richardson coefficient $c_{\lambda\mu}^{\nu}$ satisfies*

$$c_{\lambda\mu}^{\nu} \neq 0 \quad \Longleftrightarrow \quad c_{n\lambda,n\mu}^{n\nu} \neq 0.$$

In hive polytope language, the Saturation Theorem can be restated as

$$\#\left(H_{\lambda\mu}^{\nu} \cap \mathbb{Z}^d\right) \neq \varnothing \quad \Longleftrightarrow \quad \#\left(H_{n\lambda,n\mu}^{n\nu} \cap \mathbb{Z}^d\right) \neq \varnothing,$$

where $d = \binom{r+2}{2}$. The definition of hive polytopes (see Definition 2.3 above) implies that $H_{n\lambda,n\mu}^{n\nu} = n\, H_{\lambda\mu}^{\nu}$, so the Saturation Theorem is equivalent to the existence of a lattice point inside each hive polytope $H_{\lambda\mu}^{\nu}$. Now we would like to state two conjectures that generalize this theorem.

	λ, μ, ν	$C^{\mu}_{\lambda\nu}$	`LattE` runtime	`LiE` runtime
B_3	(46,42,38), (38,36,42), (41,36,44)	354440672	0m09.58s	1m45.27s
	(46,42,41), (14,58,17), (50,54,38)	88429965	0m06.38s	3m16.01s
	(15,60,67), (58,70,52), (57,38,63)	626863031	0m07.14s	6m01.43s
	(5567,2146,6241), (6932,1819,8227), (3538,4733,3648)	215676881876569849679	0m7.07s	n/a
C_3	(25,42,22), (36,38,50), (31,33,48)	87348857	0m07.48s	0m17.21s
	(34,56,36), (44,51,49), (37,51,54)	606746767	0m08.42s	2m57.27s
	(39,64,58), (65,15,72), (70,41,44)	519379044	0m07.63s	8m00.35s
	(5046,5267,7266), (7091,3228,9528), (9655,7698,2728)	1578943284716032240384	0m07.66s	n/a
D_4	(13,20,10,14), (10,20,13,20), (5,11,15,18)	41336415	2m46.88s	0m12.29s
	(12,22,9,30),(28,14,15,26),(10,24,10,26)	322610723	3m04.31s	7m03.44s
	(37,16,31,29),(40,18,35,41),(36,27,19,37)	18538329184	4m29.63s	>60m
	(2883,8198,3874,5423),(1901,9609,889,4288),(5284,9031,2959,5527)	1891293256704574356565149344	2m06.42s	n/a

Figure 4: A sample comparison of running times between `LattE` and `LiE`

4.1 First Conjecture

To show that every hive polytope contains an integral point, Knutson and Tao actually proved that every hive polytope contains an integral *vertex*. Our idea was to take a different approach to show a generalization of this last result using the basic theory of triangulations of semigroups. To develop this idea, observe that the boundary equalities and rhombus inequalities that define a hive polytope may be expressed as the set of solutions to a system of matrix equalities and inequalities:

$$H_{\lambda\mu}^{\nu} = \left\{ h \in \mathbb{R}^{(r+1)(r+2)/2} : \begin{array}{l} Bh = b(\lambda,\mu,\nu), \\ Rh \leq 0 \end{array} \right\}, \tag{2}$$

where B and R are integral matrices (depending on r), and $b(\lambda, \mu, \nu)$ is a integral vector depending on λ, μ, and ν. Here we think of a hive pattern h as a column vector of dimension $(r+1)(r+2)/2$. Note that there is some degree of choice in how the boundary equalities and rhombus inequalities are encoded as matrices B and R, respectively. However, all such encodings are equivalent for our purposes.

A polytope defined by such a system of equalities and inequalities may be *homogenized* by adding "slack variables". This produces an equivalent polytope defined as the set of nonnegative solutions to a system linear equalities. Following this procedure, we define the *homogenized hive polytope* $\tilde{H}_{\lambda\mu}^{\nu}$ by

$$\tilde{H}_{\lambda,\mu}^{\nu} = \left\{ \tilde{h} : \begin{bmatrix} B & 0 \\ R & I \end{bmatrix} \tilde{h} = \begin{bmatrix} b(\lambda,\mu,\nu) \\ 0 \end{bmatrix}, \quad \tilde{h} \geq 0 \right\}$$

(where I is the identity matrix). The equivalence between $H_{\lambda\mu}^{\nu}$ and $\tilde{H}_{\lambda\mu}^{\nu}$ is given by the linear map

$$h \mapsto \begin{bmatrix} h \\ -Rh \end{bmatrix}.$$

Note that this linear map preserves vertices and integrality. Therefore, to prove the Saturation Theorem, it suffices to show that every homogenized hive polytope contains an integral vertex. Proceeding with this idea, we make the following definitions.

Definition 4.2. Fix $r \in \mathbb{Z}$. Define the *homogenized hive matrix* to be

$$M = \begin{bmatrix} B & 0 \\ R & I \end{bmatrix}$$

(where B and R are as in equation (2)). Given an integral vector b with dimension equal to the number of rows in M, define the *generalized hive polytope* or *g-hive polytope* H_b by

$$H_b = \left\{ \tilde{h} : M\tilde{h} = b, \, \tilde{h} \geq 0 \right\}. \tag{3}$$

Note that the homogenized hive polytopes are g-hive polytopes which satisfy additional restrictions to the right-hand side vector (such has the final entries of b being 0).

We now state some very basic facts about vertices of polyhedra expressed in the form $\{x : Ax = b, x \geq 0\}$. Let a finite collection of integral points $\{a_1, \ldots, a_n\} \subset \mathbb{Z}^m$ be given, and let A be the matrix with columns $a_1, \ldots, a_n$. Define $\operatorname{cone} A$ to be the cone in $\mathbb{R}^m$ generated by the point-set $\{a_1, \ldots, a_n\}$:

$$\operatorname{cone} A = \{x_1 a_1 + \cdots + x_n a_n : x_1, \ldots, x_n \geq 0\}.$$

Then, for each vector $b \in \mathbb{Z}^m$, we have a polytope

$$P_b = \{x : Ax = b, x \geq 0\},$$

and $P_b \neq \varnothing$ if and only if $b \in \operatorname{cone} A$. In other words, there is a correspondence between nonempty polytopes P_b, $b \in \mathbb{Z}^m$, and the elements of the semigroup generated by the columns of A. The crucial property for our purposes is the following.

Lemma 4.3. *If $b \in (\operatorname{cone} A') \cap \mathbb{Z}^m$ for some $m \times m$ submatrix A' of A with $\det A' = \pm 1$, then P_b has an integral vertex.*

Proof. Suppose that $b \in (\operatorname{cone} A') \cap \mathbb{Z}^m$ for some $m \times m$ submatrix A' of A with $\det A' = \pm 1$.

Let the columns of A' be $a_{i_1}, \ldots, a_{i_m}$, and let $J = \{i_1, \ldots, i_m\}$ be the indices of these columns. Then there is a vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n_{\geq 0}$ such that $Ax = b$ and $x_i = 0$ for each $i \notin J$. Letting $x' = (x_{i_1}, \ldots, x_{i_m})$ and using Crammer's rule to solve for x' in $A'x' = b$, we find that x is an integral vector. Thus, x is an integral lattice point in the polytope P_b.

To see that x is in fact a vertex of P_b, Recall that the codimension (with respect to the ambient space) of the face containing a solution to a system of linear equalities and inequalities is the number of linearly independent equalities or inequalities satisfied with *equality*. Observe that x is a solution to the system of n equalities

$$\begin{cases} Ax = b \\ x_i = 0, \quad i \notin J. \end{cases}$$

We claim that this is a linearly independent system of equalities. For suppose otherwise. Then the zero vector is a nontrivial linear combination of the rows of A and the row-vectors e_i, $i \notin J$. But this implies that the zero vector is a nontrivial linear combination of the rows of A', which is impossible because $\det A' \neq 0$.

Thus, having shown that x satisfies the n linearly independent equalities above, we have shown that x lies in a codimension-n face of P_b, *i.e.*, x is a vertex. $\square$

We say that $a_{i_1}, \dots, a_{i_m}$ is a *unimodular* subset if the submatrix A' of A with columns $a_{i_1}, \dots, a_{i_m}$ satisfies $\det A' = \pm 1$. We say that the matrix A has a *unimodular cover* (resp. *unimodular triangulation*) if the point set $\{a_1, \dots, a_n\}$ has a unimodular cover (resp. unimodular triangulation).

Corollary 4.4. *If A has a unimodular cover, then P_b has an integral vertex for every integral $b \in \operatorname{cone}(A)$.*

Our conjecture is that this corollary applies to the homogenized hive matrix. More precisely, we conjecture the following.

Conjecture 4.5. *For each r, the homogenized hive matrix has a unimodular triangulation. Consequently, every g-hive polytope has an integral vertex.*

Since the hive polytopes are special cases of the g-hive polytopes, Conjecture 4.5 generalizes the Saturation Theorem.

Theorem 4.6. *Conjecture (4.5) is true for $r \le 6$.*

To compute the unimodular triangulations that provide a proof of Theorem 4.6 we used the software `topcom` [29]. It may be worth noting that the triangulations used to prove Theorem 4.6 were all placing triangulations.

4.2 Second Conjecture

For our second conjecture, we looked at general properties satisfied by the tensor product multiplicities for semisimple Lie Algebras of types B_r, C_r, and D_r under the operation of *stretching of multiplicities*. By *stretching of multiplicities*, we refer to the function $e \colon \mathbb{Z}_{>0} \to Z_{\ge 0}$ defined by $e(n) = C_{n\lambda,n\mu}^{n\nu}$.

It follows from the definitions of the BZ-polytopes that, given any highest weights λ, μ, ν of a semisimple Lie algebra, $C_{n\lambda,n\mu}^{n\nu} = e(n)$ is a quasi-polynomial in n. Indeed, $e(n)$ is, in polyhedral language, the *Ehrhart quasi-polynomial* of the corresponding BZ-polytope. We recall the basic theory of Ehrhart quasi-polynomials. Its origins can be traced to the work of Ehrhart [15] in the 1960's (see Chapter 4 of [31] for an excellent introduction).

Given a convex polytope P, let

$$nP = \{x : (1/n)x \in P\}, \qquad n = 1, 2, \dots .$$

If P is a d-dimensional rational polytope in $\mathbb{R}^k$, then the counting function $i_P(n) = \#(nP \cap \mathbb{Z}^k)$ is a quasi-polynomial function of degree d; that is, there are polynomials $f_1(n), \dots, f_N(n)$ of degree d s.t.

$$i_P(n) = \begin{cases} f_1(n) & \text{if } n \equiv 1 \mod N, \\ \vdots & \\ f_N(n) & \text{if } n \equiv N \mod N. \end{cases}$$

If we put $P = H^{\nu}_{\lambda\mu}$, then the Ehrhart quasi-polynomial of P is just the stretched Littlewood–Richardson coefficient $c^{n\nu}_{n\lambda,n\mu}$. The Ehrhart quasi-polynomials of hive polytopes have been studied by several authors. Our experiments support the conjecture of King et al [18] that the coefficients are in fact positive. Since large weights can be computed with lattice point enumeration, it is possible to produce the Ehrhart quasi-polynomials for the stretched Clebsch–Gordan coefficients in the other types. See Figures 5–7 for some sample examples out of the many hundreds generated. Our experiments motivate the following "stretching conjecture".

Conjecture 4.7. *(Stretching Conjecture) Given highest weights λ, μ, ν of a Lie algebra of type A_r, B_r, C_r, or D_r, let*

$$C^{n\nu}_{n\lambda,n\mu} = \begin{cases} f_1(n) & \textit{if } n \equiv 1 \quad \textit{mod}\, N, \\ \vdots & \\ f_N(n) & \textit{if } n \equiv N \quad \textit{mod}\, N \end{cases}$$

be the quasi-polynomial representation of the stretched Clebsch–Gordan coefficient $C^{n\nu}_{n\lambda,n\mu}$. Then the coefficients of each polynomial f_i are all nonnegative.

The type A_r case of this conjecture was made by King, Tollu, and Toumazet in [18]. That Conjecture 4.7 implies the Saturation Theorem follows from a result of Derksen and Weyman [11] showing that the Ehrhart quasi-polynomials of Hive polytopes are in fact just polynomials.

We should remark that the saturation property is known not to hold in the root systems B_r, C_r, and D_r. A simple example in B_2, due to Kapovich, Leeb, and Millson [17], is given by setting $\lambda = \mu = \nu = (1, 0)$ (with respect to the basis of fundamental weights). In this case we have

$$C^{n\nu}_{n\lambda,n\mu} = \begin{cases} 0 & \text{if } n \text{ is odd,} \\ 1 & \text{if } n \text{ is even.} \end{cases}$$

This example also demonstrates why the Stretching Conjecture is not contradicted by the failure of the saturation property in the root systems B_r, C_r, or D_r. Since the stretched multiplicities are not necessarily polynomials in these cases, it is possible for them to evaluate to zero for some nonnegative integer while still having all nonnegative coefficients.

Acknowledgements: We are thankful to Arkady Berenstein, Charles Cochet, Misha Kapovich, Allen Knutson, Peter Littelmann, Ezra Miller, Hariharan Narayanan, Jörg Rambau, Francisco Santos, Anna Schilling, Etienne Rassart, Michèle Vergne, and Andrei Zelevinsky for useful comments.

References

[1] W. Baldoni-Silva, J.A. De Loera, and M. Vergne, Counting integer flows in networks, *Foundations of Comp. Math.* Vol. 4, No. 3, 2004, 277-314.

[2] W. Baldoni-Silva, M. Beck, C. Cochet, and M. Vergne, partition functions for classical root systems, manuscript 2005.

[3] A.I. Barvinok, A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.* **19** (1994), no. 4, 769–779.

[4] A.D. Berenstein and A.N. Kirillov, Groups generated by involutions, Gelfand–Tsetlin patterns, and combinatorics of Young tableaux, *Algebra i Analiz* 7 (1995), no. 1, 92–152 (Russian). Translation in *St. Petersburg Math.* J. 7 (1996), no. 1, 77–127.

[5] A.D. Berenstein and A.V. Zelevinsky, Triple multiplicities for sl$(r+1)$ and the spectrum of the exterior algebra of the adjoint representation. *J. Algebraic Combin.* **1** (1992), no. 1, 7–22.

[6] A.D. Berenstein and A.V. Zelevinsky, Tensor product multiplicities, canonical bases and totally positive varieties. *Invent. Math.* **143** (2001), no. 1, 77–128.

[7] A.S. Buch, The saturation conjecture (after A. Knutson and T. Tao). With an appendix by William Fulton. *Enseign. Math. (2)* **46** (2000), no. 1-2, 43–60.

[8] C. Cochet, Kostka numbers and Littlewood–Richardson coefficients, to appear in *Contemporary Math.* proceedings of the conference "Integral Points in Rational Polytopes", Snowbird, Utah July 2003.

[9] C. Cochet, Vector partition function and representation theory, manuscript 2005. Paper and software available from `http://www.math.jussieu.fr/~cochet/`

[10] A.M. Cohen and W.A. de Graaf, Lie algebraic computation *Comp. Phys. Commun.* 97:53–62 (1996).

[11] H. Derksen and J. Weyman, On the Littlewood–Richardson polynomials, *J. of Algebra*. **255** (2002), no. 2, 247–257.

[12] J.A. De Loera and R. Hemmecke and J. Tauzer and R. Yoshida, Effective lattice point counting in rational convex polytopes, *J. Symbolic Comp.* vol. 38 no. 4 (2004), 1273–1302.

[13] J.A. De Loera, D. Haws, R. Hemmecke, P. Huggins, J. Tauzer and R. Yoshida, A user's guide for LattE v.1.1, 2003, software and manual available at `www.math.ucdavis.edu/~latte`.

λ, μ, ν	$C^{n\mu}_{n\lambda, n\nu}$	`LattE` runtime
(0, 15, 5) (12, 15, 3) (6, 15, 6)	$\begin{cases} \frac{68339}{64}\,n^5 + \frac{407513}{384}\,n^4 + \frac{13405}{32}\,n^3 + \frac{9499}{96}\,n^2 + \frac{107}{8}\,n + 1, & n \text{ even} \\ \frac{68339}{64}\,n^5 + \frac{407513}{384}\,n^4 + \frac{13405}{32}\,n^3 + \frac{16355}{192}\,n^2 + \frac{659}{64}\,n + \frac{75}{128}, & n \text{ odd} \end{cases}$	7m08.57s
(4, 8, 11) (3, 15, 10) (10, 1, 3)	$\begin{cases} \frac{13}{4}\,n^2 + 3\,n + 1, & n \text{ even} \\ \frac{13}{4}\,n^2 + 3\,n + 3/4, & n \text{ odd} \end{cases}$	0m01.64s
(8, 1, 3) (11, 13, 3) (8, 6, 14)	$\begin{cases} \frac{121}{576}\,n^6 + \frac{1129}{640}\,n^5 + \frac{6809}{1152}\,n^4 + \frac{163}{16}\,n^3 + \frac{2771}{288}\,n^2 + \frac{191}{40}\,n + 1, & n \text{ even} \\ \frac{121}{576}\,n^6 + \frac{1129}{640}\,n^5 + \frac{6809}{1152}\,n^4 + \frac{1933}{192}\,n^3 + \frac{659}{72}\,n^2 + \frac{8003}{1920}\,n + \frac{93}{128}, & n \text{ odd} \end{cases}$	0m10.26s
(8, 9, 14) (8, 4, 5) (1, 5, 15)	$\begin{cases} \frac{4117}{192}\,n^6 + \frac{50369}{640}\,n^5 + \frac{14829}{128}\,n^4 + \frac{703}{8}\,n^3 + \frac{3541}{96}\,n^2 + \frac{341}{40}\,n + 1, & n \text{ even} \\ \frac{4117}{192}\,n^6 + \frac{50369}{640}\,n^5 + \frac{14829}{128}\,n^4 + \frac{5599}{64}\,n^3 + \frac{3451}{96}\,n^2 + \frac{5001}{640}\,n + \frac{97}{128}, & n \text{ odd} \end{cases}$	0m13.29s
(10, 5, 6) (5, 4, 10) (0, 7, 12)	$\begin{cases} \frac{669989}{960}\,n^5 + \frac{286355}{384}\,n^4 + \frac{10803}{32}\,n^3 + \frac{7993}{96}\,n^2 + \frac{1427}{120}\,n + 1, & n \text{ even} \\ \frac{669989}{960}\,n^5 + \frac{286355}{384}\,n^4 + \frac{10803}{32}\,n^3 + \frac{15509}{192}\,n^2 + \frac{10081}{960}\,n + \frac{65}{128}, & n \text{ odd} \end{cases}$	2m52.39s

Figure 5: Stretched Clebsch–Gordan coefficients for B_3.

λ, μ, ν	$C_{n\lambda,n\nu}^{n\mu}$	LattE runtime
(1,13,6) (14,15,5) (5,11,7)	$\begin{cases} \frac{5937739}{5760}\,n^6 + \frac{87023}{40}\,n^5 + \frac{936097}{576}\,n^4 + \frac{27961}{48}\,n^3 + \frac{85397}{720}\,n^2 + \frac{883}{60}\,n + 1, & n \text{ even} \\ \frac{5937739}{5760}\,n^6 + \frac{87023}{40}\,n^5 + \frac{936097}{576}\,n^4 + \frac{27961}{48}\,n^3 + \frac{657931}{5760}\,n^2 + \frac{3097}{240}\,n + 3/4, & n \text{ odd} \end{cases}$	21m20.59s
(4,15,14) (12,12,10) (4,9,8)	$\begin{cases} \frac{22199219}{2880}\,n^6 + \frac{8154617}{960}\,n^5 + \frac{4500665}{1152}\,n^4 + \frac{31297}{32}\,n^3 + \frac{226903}{1440}\,n^2 + \frac{2021}{120}\,n + 1, & n \text{ even} \\ \frac{22199219}{2880}\,n^6 + \frac{8154617}{960}\,n^5 + \frac{4500665}{1152}\,n^4 + \frac{31297}{32}\,n^3 + \frac{217363}{1440}\,n^2 + \frac{13513}{960}\,n + \frac{85}{128}, & n \text{ odd} \end{cases}$	17m05.74s
(9,0,8) (8,12,9) (7,7,3)	$1/30\,n^5 + 3/8\,n^4 + \frac{19}{12}\,n^3 + \frac{25}{8}\,n^2 + \frac{173}{60}\,n + 1$	0m00.61s
(10,2,7) (8,10,1) (7,5,5)	$\begin{cases} \frac{596153}{1152}\,n^6 + \frac{53425}{48}\,n^5 + \frac{502621}{576}\,n^4 + \frac{5577}{16}\,n^3 + \frac{11941}{144}\,n^2 + \frac{149}{12}\,n + 1, & n \text{ even} \\ \frac{596153}{1152}\,n^6 + \frac{53425}{48}\,n^5 + \frac{502621}{576}\,n^4 + \frac{5577}{16}\,n^3 + \frac{94097}{1152}\,n^2 + \frac{131}{12}\,n + \frac{23}{32}, & n \text{ odd} \end{cases}$	19m24.55s
(10,10,15) (11,3,15) (10,7,15)	$\begin{cases} \frac{6084163}{320}\,n^6 + \frac{507527}{30}\,n^5 + \frac{1185853}{192}\,n^4 + \frac{59995}{48}\,n^3 + \frac{43039}{240}\,n^2 + \frac{357}{20}\,n + 1, & n \text{ even} \\ \frac{6084163}{320}\,n^6 + \frac{507527}{30}\,n^5 + \frac{1185853}{192}\,n^4 + \frac{59995}{48}\,n^3 + \frac{144751}{960}\,n^2 + \frac{883}{80}\,n + \frac{25}{64}, & n \text{ odd} \end{cases}$	16m05.08s

Figure 6: Stretched Clebsch–Gordan coefficients for C_3.

λ, μ, ν	$C_{n\lambda,n\nu}^{n\mu}$	`LattE` runtime
(0, 2, 10, 5) (4, 11, 9, 11) (5, 8, 6, 9)	$\begin{cases} \frac{625007}{10080}\,n^7 + \frac{729157}{2880}\,n^6 + \frac{77197}{180}\,n^5 + \frac{449539}{1152}\,n^4 + \frac{298979}{1440}\,n^3 + \frac{95189}{1440}\,n^2 + \frac{10079}{840}\,n + 1, & n \text{ even} \\ \frac{625007}{10080}\,n^7 + \frac{729157}{2880}\,n^6 + \frac{77197}{180}\,n^5 + \frac{449539}{1152}\,n^4 + \frac{298979}{1440}\,n^3 + \frac{95189}{1440}\,n^2 + \frac{10079}{840}\,n + \frac{127}{128}, & n \text{ odd} \end{cases}$	20m24.79s
(2, 7, 12, 2) (11, 10, 5, 9) (13, 11, 1, 1)	$\begin{cases} \frac{34675903}{80640}\,n^8 + \frac{3037051}{1680}\,n^7 + \frac{9121453}{2880}\,n^6 + \frac{241181}{80}\,n^5 + \frac{615083}{360}\,n^4 + \frac{8947}{15}\,n^3 + \frac{107791}{840}\,n^2 + \frac{6721}{420}\,n + 1 & n \text{ even} \\ \frac{34675903}{80640}\,n^8 + \frac{3037051}{1680}\,n^7 + \frac{9121453}{2880}\,n^6 + \frac{241181}{80}\,n^5 + \frac{615083}{360}\,n^4 + \frac{8947}{15}\,n^3 + \frac{107791}{840}\,n^2 + \frac{6721}{420}\,n + \frac{239}{256} & n \text{ odd} \end{cases}$	123m59.76s
(3, 11, 0, 10) (2, 15, 10, 15) (10, 12, 11, 0)	$\begin{cases} \frac{53609}{60}\,n^6 + \frac{25631}{15}\,n^5 + \frac{63779}{48}\,n^4 + \frac{1627}{3}\,n^3 + \frac{2497}{20}\,n^2 + \frac{239}{15}\,n + 1, & n \text{ even} \\ \frac{53609}{60}\,n^6 + \frac{25631}{15}\,n^5 + \frac{63779}{48}\,n^4 + \frac{1627}{3}\,n^3 + \frac{2497}{20}\,n^2 + \frac{239}{15}\,n + \frac{15}{16}, & n \text{ odd} \end{cases}$	2m37.73s
(10, 1, 12, 4) (1, 12, 0, 3) (0, 5, 3, 4)	$5\,n^2 + 4\,n + 1$	0m01.63s
(12, 2, 5, 13) (15, 6, 10, 11) (2, 0, 12, 13)	$\begin{cases} \frac{455263}{2016}\,n^7 + \frac{447281}{576}\,n^6 + \frac{198433}{180}\,n^5 + \frac{971011}{1152}\,n^4 + \frac{108787}{288}\,n^3 + \frac{28969}{288}\,n^2 + \frac{12631}{840}\,n + 1, & n \text{ even} \\ \frac{455263}{2016}\,n^7 + \frac{447281}{576}\,n^6 + \frac{198433}{180}\,n^5 + \frac{971011}{1152}\,n^4 + \frac{108787}{288}\,n^3 + \frac{28969}{288}\,n^2 + \frac{12631}{840}\,n + \frac{127}{128}, & n \text{ odd} \end{cases}$	4m25.90s

Figure 7: Stretched Clebsch–Gordan coefficients for D_4.

[14] P. Diaconis and A. Gangolli, Rectangular arrays with fixed margins. *Discrete probability and algorithms (Minneapolis, MN, 1993)*, 15–41, IMA Vol. Math. Appl., 72, Springer, New York, 1995. ISBN 0-387-94532-6

[15] E. Ehrhart, *Polynômes arithmétiques et méthode des polyèdres en combinatoire*. International Series of Numerical Mathematics, Vol. 35. Birkhäuser Verlag, Basel-Stuttgart, 1977.

[16] J.E. Humphreys, *Introduction to Lie algebras and representation theory*, Springer, New York, 1972.

[17] M. Kapovich, B. Leeb, and J. Millson, The generalized triangle inequalities in symmetric spaces and buildings with applications to algebra, arXiv:math.RT/0210256

[18] R.C. King, C. Tollu, and F. Toumazet, Stretched Littlewood–Richardson and Kostka coefficients, to appear in *CRM Proceedings and Lecture Notes, Montréal*, Vol 34, 2003.

[19] A. Knutson and T. Tao, The honeycomb model of $GL_n(\mathbb{C})$ tensor products I: Proof of the Saturation Conjecture, *J. Amer. Math. Soc.* **12** (1999), 1055–1090, arXiv:math.RT/9807160.

[20] A. Knutson, T. Tao, and C. Woodward, A positive proof of the Littlewood–Richardson rule using the octahedron recurrence. *Electron. J. Combin.*, to appear, arXiv:math.CO/0306274.

[21] W. Fulton and J. Harris, *Representation Theory: A first course*, Graduate texts in Mathematics vol 129, Springer, New York, 1991.

[22] W. Fulton, Young Tableaux, vol. 35, London Math. Soc. Student texts. Cambridge Press 1997.

[23] M. Kashiwara, Crystalizing the q-analogue of universal enveloping algebras, *Commun. Math. Phys.* 133, (1990), 249–260.

[24] P. Littelmann, Cones, crystals and patterns, *Transf. Groups* 3, (1998), 145–179.

[25] Maple, `http://www.maplesoft.com/`

[26] K. Mulmuley and M. Sohoni, Geometric complexity theory, P vs. NP and explicit obstructions. *Advances in algebra and geometry (Hyderabad, 2001)*, 239–261, *Hindustan Book Agency, New Delhi*, 2003. Available at `http://ramakrishnadas.cs.uchicago.edu/hyderabad.ps`

[27] H. Narayanan, The computation of Kostka numbers and Littlewood–Richardson coefficients is $\#P$-complete, preprint (2005), arXiv:math.CO/0501176.

[28] I. Pak and E. Vallejo, Combinatorics and geometry of Littlewood–Richardson cones, preprint (2003), to appear in *Europ. J. Combinatorics*, arXiv:math.CO/0407170.

[29] J. Rambau, TOPCOM: Triangulations of Point Configurations and Oriented Matroids, in *Proceedings of the International Congress of Mathematical Software ICMS 2002.* to appear. Software available at `http://www.zib.de/rambau/TOPCOM/`

[30] A. Schrijver, Theory of Linear and Integer Programming, John Wiley & Sons Ltd., 1986.

[31] R.P. Stanley, Enumerative Combinatorics, Volume I, Cambridge University Press, 1997.

[32] R.P. Stanley, Enumerative Combinatorics, Volume II, Cambridge University Press, 1999.

[33] R.P. Stanley, Combinatorial aspects of the Schubert calculus, in *Combinatoire et représentation du groupe symétrique* 217–251, Lecture Notes in Math., Vol 579, Springer, Berlin 1977.

[34] J.A. Stembridge *Coxeter and Weyl*, `http://www.math.lsa.umich.edu/~jrs/maple.html`

[35] J.A. Stembridge, Computational aspects of root systems, Coxeter groups, and Weyl characters, in *Interactions of Combinatorics and Representation Theory*, MSJ Memoirs 11, Math. Soc. Japan, Tokyo, 2001, 1–38.

[36] M.A.A. van Leeuwen, LiE a software package for Lie group computations, *Euromath Bull.*, 1, 1994, 83–94. Software and manual available at `http://wwwmathlabo.univ-poitiers.fr/~maavl/LiE/`

[37] B.G. Wybourne, The Littlewood–Richardson Rule - The Cornerstone for Computing Group Properties, in Proc. Stefan Banach Int. Math. Centre Seminar on Combinatorics vol. 26, (1988), 475–482.