

```

gap> g:= SymmetricGroup( 4 );
Sym( [ 1 .. 4 ] )
gap> tbl:= CharacterTable( g ); HasIrr( tbl );
i5 : betti(t,Weights=>{1,0})
false
      0 1 2 3 4
o5 = total: 1 4 13 14 4
      0: 1 . . .
      1: . 2 2 4 2
      2: . 2 5 6 .
      3: . . 4 . 2
      4: . . . 4 .
      5: . . 2 . .
gap> tblmod2:= CharacterTable( tbl, 2 );
      BrauerTable( Sym( [ 1 .. 4 ] ), 2 )
gap> tblmod2 = CharacterTable( tbl, 2 );
true
gap> tblmod2 = BrauerTable( tbl, 2 );
true
o5 : BrauerTable
i6 : betti(t,Weights=>{0,1})
      0 1 2 3 4
o6 = total: 1 4 13 14 4
      0: 1 . . .
      1: . 2 2 4 2
      2: . 2 5 6 .
      3: . . 4 . 2
      4: . . . 4 .
      5: . . 2 . .
gap> libtbl:= CharacterTable( "M" );
      CharacterTable( "M" )
gap> CharacterTableRegular( libtbl, 2 );
      BrauerTable( "M" )
gap> BrauerTable( libtbl, 2 );
fail
gap> CharacterTable( "Symmetric", 4 );
      CharacterTable( "Sym(4)" )
i7 : t1 = betti(t,Weights=>{1,1})
gap> ComputedBrauerTables( tbl );
      [ , BrauerTable( Sym( [ 1 .. 4 ] ), 2 ), ]
      0 1 2 3 4
o7 = total: 1 4 13 14 4
      0: 1 . . .
      1: . . . .
      2: . . . .
      3: . 2 . .
      4: . . . .
      5: . 2 . .
      6: . . 1 .
      7: . . 8 6 .
      8: . . 4 8 4
      ring r1 = 32003,(x,y,z),ds;
      int a,b,c,t=11,5,3,0;
      poly f = x^a*y^b+z^(3*c)+x^(c+2)*y^(c-1)+x^(
      x^(c-2)*y^c*(y^2+t*x)^2;
      option(noprot);
      timer=1;
      ring r2 = 32003,(x,y,z),dp;
      poly f=imap(r1,f);
      ideal j=jacob(f);
      vdim(std(j));
==> 536
      vdim(std(j+f));
==> 195
      timer=0; // reset timer
o7 : BettiTally
i8 : peek t1
o8 = BettiTally{ (0, {0, 0}, 0) => 1 }
      (1, {2, 2}, 4) => 2
      (1, {3, 3}, 6) => 2
      (2, {3, 7}, 10) => 2
      (2, {4, 4}, 8) => 1
      (2, {4, 5}, 9) => 4
      (2, {5, 4}, 9) => 4
      (2, {7, 3}, 10) => 2
      (3, {4, 7}, 11) => 4
      (3, {5, 5}, 10) => 6
      (3, {7, 4}, 11) => 4
      (4, {5, 7}, 12) => 2
      (4, {7, 5}, 12) => 2

```


Software for multiplier ideals

ZACH TEITLER

ABSTRACT: We describe a new software package for computing multiplier ideals in certain cases, including monomial ideals, monomial curves, generic determinantal ideals, and hyperplane arrangements. In these cases we take advantage of combinatorial formulas for multiplier ideals given by results of Howald, Thompson, and Johnson. The package uses the package *Normaliz*. It is available as a library for *Macaulay2*.

INTRODUCTION. Multiplier ideals have been applied to a number of problems in algebraic geometry in recent years, most spectacularly in recent major advances in the minimal model program [Hacon and McKernan 2007; Birkar et al. 2010] that built on earlier work showing the deformation invariance of plurigena [Siu 1998]. Other applications include several results on singularities and linear series [Lazarsfeld 2004; Ein and Mustață 2006], a bound for symbolic powers [Ein et al. 2001], and applications to algebraic statistics [Watanabe 2009; Zwiernik 2011; Drton et al. 2009, Chapter 5]. New applications of multiplier ideals continue to emerge in topics such as Chow stability [Lee 2008] and singularities in generic liaison [Niu 2014]. With broad and growing interest in multiplier ideals, it is increasingly valuable to compute examples.

For a thorough introduction to multiplier ideals see [Lazarsfeld 2004]. Here is a definition of multiplier ideals in terms of resolution of singularities: Suppose X is a smooth variety over a field \mathbb{k} (we may assume X is affine, or even just \mathbb{k}^n , since we are primarily interested in local issues), $I \subset \mathcal{O}_X$ is a nonzero ideal sheaf, and $\mu : Y \rightarrow X$ is a log resolution of I , so that the total transform $I\mathcal{O}_Y$ defines a divisor F with simple normal crossings support, $F = \sum a_i E_i$, where the E_i are distinct reduced components of F . Then for each real number $c \geq 0$ the c -th multiplier ideal is defined by

$$\mathcal{J}(I^c) = \mu_* \mathcal{O}_Y(K_{Y/X} - \lfloor c \cdot F \rfloor),$$

where $K_{Y/X}$ is the relative canonical divisor of Y over X , defined locally by the vanishing of the determinant of the Jacobian $d\mu$, and $\lfloor c \cdot F \rfloor$ denotes the component-wise round-down of the \mathbb{R} -divisor $c \cdot F$, given by $\lfloor c \cdot F \rfloor = \sum \lfloor ca_i \rfloor E_i$.

MSC2010: primary 14Q99, 14F18; secondary 13A15, 13P25.

Keywords: multiplier ideal, log canonical threshold, jumping number.

In theory it is algorithmic to compute multiplier ideals by computing a resolution of singularities of I followed by a sheaf pushforward. In practice it is more difficult; see [Frühbis-Krüger 2014].

Shibuta’s [2011] algorithm for computing Bernstein–Sato polynomials and multiplier ideals via Gröbner basis methods in Weyl algebras (which he implemented in *Risa/Asir*) was refined and implemented in the *Dmodules* library for *Macaulay2* by Berkesch and Leykin [2010]. The *Dmodules* library can compute multiplier ideals and jumping numbers of arbitrary ideals, but due to the difficulty of the computations can only handle modestly sized examples.

We describe a new software package named *MultiplierIdeals* that computes multiplier ideals of special ideals, including monomial ideals, ideals of monomial curves, generic determinantal ideals, and hyperplane arrangements, via combinatorial methods, using the *Normaliz* software and interface to *Macaulay2* by Bruns, Ichim, and Kämpf [Bruns and Ichim 2010; Bruns and Kämpf 2010]. The combinatorial methods allow computations of somewhat larger examples than can be handled by general methods.

Wherever possible we work over an arbitrary field \mathbb{k} . Since multiplier ideals in our cases are computed by resolutions defined over \mathbb{Z} (or over the \mathbb{Z} -algebra generated by the coefficients of the defining equations of the input data), we may work in arbitrary characteristic.

Our package also computes certain quantities associated to multiplier ideals: the log canonical thresholds and jumping numbers. Because of the round-down operation, $\mathcal{J}(I^{c+\epsilon}) = \mathcal{J}(I^c)$ for sufficiently small $\epsilon > 0$. A real number $c \geq 0$ is a jumping number of I if $\mathcal{J}(I^c) \neq \mathcal{J}(I^{c-\epsilon})$ for all $\epsilon > 0$. Every jumping number is in fact rational. The smallest strictly positive jumping number is called the log canonical threshold of I , denoted $\text{lct}(I)$. It turns out that $\mathcal{J}(I^0) = (1)$ is the trivial ideal, so $\text{lct}(I)$ is the supremum of c such that $\mathcal{J}(I^c) = (1)$; equivalently, $\text{lct}(I)$ is the first value of c such that $\mathcal{J}(I^c) \neq (1)$.

The portion dealing with monomial ideals was written first and distributed as the package *MonomialMultiplierIdeals*. The portion dealing with monomial curves was written by C. Raicu, B. Snapp, and the author at the 2011 IMA Special Workshop on *Macaulay2*, and distributed as the package *SpaceCurvesMultiplierIdeals*. The portion dealing with hyperplane arrangements is based on code written by Denham and Smith for the *HyperplaneArrangements* package [2011]. These portions were all integrated into the present package, and computations with generic determinantal ideals added, at the 2012 *Macaulay2* Workshop at Wake Forest.

MONOMIAL IDEALS. For a monomial ideal $I \subset \mathbb{k}[x_1, \dots, x_n]$, let $\text{monom}(I) \subset \mathbb{Z}_{\geq 0}^n$ be the set of exponent vectors of monomials in I . The Newton polyhedron $\text{Newt}(I)$ is the convex hull of $\text{monom}(I)$. Let $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^n$.

Multiplier ideals of monomial ideals are described by the following theorem of Howald:

Theorem 1 [Howald 2001; Blickle 2004]. *The multiplier ideal $\mathcal{J}(I^c)$ is the monomial ideal containing x^v if and only if $v + \mathbf{1} \in \text{Int}(c \cdot \text{Newt}(I))$. Here Int denotes the topological interior of $c \cdot \text{Newt}(I)$ relative to the nonnegative orthant, that is, as a subset of $(\mathbb{R}_{\geq 0})^n$.*

In other words, the multiplier ideal is the quotient ideal

$$\mathcal{J}(I^c) = (x^v : v \in \text{Int}(c \cdot \text{Newt}(I))) : x^{\mathbf{1}}.$$

The Newton polyhedron $\text{Newt}(I)$ is defined by a system of inequalities $Av \geq b$, where A is an $r \times n$ matrix, b is a vector, and \geq is the partial order of entry-wise comparison, where $a \geq b$ if and only if $a_i \geq b_i$, $1 \leq i \leq r$. Then $c \cdot \text{Newt}(I)$ is defined by $Av \geq cb$. The interior $\text{Int}(c \cdot \text{Newt}(I))$ is the solution of the system of inequalities given by

$$\begin{cases} A_i v > cb_i & \text{if } b_i \neq 0, \\ A_i v \geq cb_i = 0 & \text{if } b_i = 0. \end{cases}$$

Since $\text{Newt}(I)$ is a rational polyhedron, we can (and do) take the A and b to have integer entries. Furthermore, since I is an ideal the entries of A and b are nonnegative. In practice, it is sufficient to compute $\mathcal{J}(I^c)$ for rational $c = p/q$, and this can be done as follows: To find the integer vectors v lying in the topological interior of the solution region to $Av \geq cb$ (equivalently, $qAv \geq pb$), we add 1 to the nonzero entries of pb , yielding a vector b' with entries $b'_i = pb_i + 1$ if $b_i \neq 0$, and $b'_i = pb_i = b_i = 0$ otherwise. Then the multiplier ideal $\mathcal{J}(I^c)$ is the quotient $(x^v : qAv \geq b') : x^{\mathbf{1}}$.

The software *Normaliz* can compute the defining inequalities $Av \geq b$ of $\text{Newt}(I)$ and the solutions to the modified system $qAv \geq b'$; *Macaulay2* can compute the ideal quotient by $x^{\mathbf{1}}$, giving the multiplier ideal:

```
Macaulay2, version 1.6
i1 : needsPackage "MultiplierIdeals";
i2 : R = QQ[x,y,z,w];
i3 : I = monomialIdeal(x*y, x*z, y*z, y*w, z*w^2);
o3 : MonomialIdeal of R
i4 : logCanonicalThreshold(I)
o4 = 2
i5 : multiplierIdeal(I,7/3)
      2
o5 = ideal (y, z*w, z , x*z)
i6 : toString jumpingNumbers(I)
o6 = {{2, 7/3, 5/2, 8/3, 3, 10/3, 7/2, 11/3, 4}, {ideal(z,y),
      ideal(y,z*w,z^2,x*z), ideal(z*w,y*w,y*z,x*z,y^2,x*y),
      ideal(y*w,y*z,x*z,y^2,x*y,z*w^2,z^2*w),
      ideal(y*z*w,y^2*w,y*z^2,x*z^2,y^2*z,x*y*z,x*y^2,z^2*w^2), ...
```

The `jumpingNumbers` command produces a list with two elements:

- (1) A list of the jumping numbers of I in the interval $(0, k(I)]$, where $k(I)$ is the analytic spread of I . A different interval may be specified as an optional argument.
- (2) A list of the multiplier ideals at the jumping numbers. (The list is truncated in the above example.)

Thus the output of the last command says that this ideal I has jumping numbers $2, 7/3, \dots$, and gives the corresponding multiplier ideals: $\mathcal{J}(I^2) = (z, y)$, $\mathcal{J}(I^{7/3}) = (y, zw, z^2, xz)$, and so on. Multiplier ideals and jumping numbers for $c > k(I)$ are given by Skoda's theorem [Lazarsfeld 2004, Theorem 9.6.21]. Namely, for $c > k(I)$, $\mathcal{J}(I^c) = I\mathcal{J}(I^{c-1})$; and $c > k(I)$ is a jumping number if and only if $c - 1$ is a jumping number.

In the above example, the log canonical threshold, single multiplier ideal $\mathcal{J}(I^{7/3})$, and list of nine jumping numbers and multiplier ideals were each computed in a fraction of a second on a 2012 MacBook with dual-core 64-bit 2.9 GHz CPU and 8 GB RAM. By way of comparison, the *Dmodules* package takes about 42 seconds to compute the log canonical threshold on the same machine, and about 84 seconds to compute $\mathcal{J}(I^{7/3})$. This comparison is only intended to illustrate the advantages of using special algorithms where available, and we remind the reader that the *DModules* package uses a general method.

For monomial ideals, extra information is available: for any monomial x^v , the package computes the threshold value $\min\{c : x^v \notin \mathcal{J}(I^c)\}$, and the list of facets of the Newton polyhedron that impose the nonmembership:

```
i7 : toString logCanonicalThreshold(I, z^2*w)
o7 = (3, matrix {{2, 2, 1, 1, -3}, {2, 2, 0, 1, -2}})
```

This output means that $z^2w \notin \mathcal{J}(I^3)$ but $z^2w \in \mathcal{J}(I^c)$ for $c < 3$. That is, for the exponent vector $v = (0, 0, 2, 1)$, $v + \mathbf{1}$ lies on the boundary of $3 \cdot \text{Newt}(I)$; and furthermore it lies on the intersection of two facets, the ones scaled up from the facets of $\text{Newt}(I)$ defined by $2x + 2y + z + w = 3$ and $2x + 2y + w = 2$.

The log canonical threshold of the ideal I itself is the threshold value for $1 = x^0$.

MONOMIAL CURVES. An affine monomial curve is one parametrized by $t \mapsto (t^{a_1}, \dots, t^{a_n})$. We can and do assume that $1 \leq a_1 \leq \dots \leq a_n$ and $\gcd(a_1, \dots, a_n) = 1$. For convenience we denote this curve by $C(a_1, \dots, a_n)$. It has a singularity at the origin when $a_1 \geq 2$. The defining ideal is the kernel of the map $\mathbb{k}[x_1, \dots, x_n] \rightarrow \mathbb{k}[t]$ given by $x_i \mapsto t^{a_i}$. This is a binomial ideal.

The multiplier ideals of affine monomial curves in dimension $n = 3$ have been found by Howard Thompson [2014],¹ using the combinatorial description of the

¹This paper states the result over \mathbb{C} , but it holds over any field.

resolution of singularities of a binomial ideal given in [González Pérez and Teissier 2002]. This yields a combinatorial formula in terms of the vector (a, b, c) of exponents appearing in the parametrization $t \mapsto (t^a, t^b, t^c)$. Our software package implements Thompson's result, again calling on *Normaliz* to find generators for the semigroup of integer solutions to certain linear inequalities:

```
i8 : R = QQ[x,y,z]; S = QQ[t];
i10 : I = kernel map(S,R,{t^3,t^4,t^5}) -- ideal of C(3,4,5)
      2      2      2      3
o10 = ideal (y - x*z, x y - z , x - y*z)
```

To compute the multiplier ideals and log canonical threshold of I , we input the list of exponents in the parametrization:

```
i11 : toString logCanonicalThreshold(R,{3,4,5})
o11 = 13/9
i12 : multiplierIdeal(R,{3,4,5},13/9)
o12 = ideal (z, y, x)
```

GENERIC DETERMINANTAL IDEALS. Let $X = (x_{i,j})_{1 \leq i \leq m, 1 \leq j \leq n}$ be an $m \times n$ generic matrix, meaning one whose entries are independent variables. Let $I_r(X)$ be the ideal generated by the $r \times r$ minors of X . The multiplier ideals of $I_r(X)$ have been found by Amanda Johnson [2003]:²

Theorem 2. *With X , m , n , and r as above, the multiplier ideals are given by the following intersection of symbolic powers of determinantal ideals:*

$$\mathcal{J}(I_r(X)^c) = \bigcap_{i=1}^r I_i(X)^{(\lfloor c(r+1-i) \rfloor + 1 - (n-i+1)(m-i+1))}.$$

Recall that symbolic powers of generic determinantal ideals may be expressed as

$$I_r(X)^{(a)} = \sum_{\kappa_1 + \dots + \kappa_s = a} \prod_{i=1}^s I_{r-1+\kappa_i}(X),$$

with the sum taken over partitions of a . See [Bruns and Vetter 1988, Theorem 10.4].

We may compute multiplier ideals of determinantal ideals in our software by giving the matrix X and the size of minors. Here we examine multiplier ideals of the size-2 and size-3 minors of a 4×5 generic matrix:

```
i13 : x = getSymbol"x"; R = QQ[x_1..x_20];
i15 : X = genericMatrix(R,4,5); -- a 4x5 generic matrix
i16 : logCanonicalThreshold(X,2) -- lct of the ideal of 2x2 minors
o16 = 10
i17 : multiplierIdeal(X,2,10) == minors(1,X) -- J(I^10) where I = 2x2 minors
o17 = true
```

²This dissertation states the result for algebraically closed fields, but it holds over any field.


```
i18 : multiplierIdeal(X,2,11) == (minors(1,X))^3 -- J(I^11)
o18 = true
```

HYPERPLANE ARRANGEMENTS. A formula for multiplier ideals of hyperplane arrangements was found by Mustață [2006] and simplified in [Teitler 2008].³ The *HyperplaneArrangements* package [Denham and Smith 2011] uses these results to compute multiplier ideals and log canonical thresholds of hyperplane arrangements. To this we add the ability to compute jumping numbers and other minor modifications. I thank Graham Denham and Gregory G. Smith, the authors of *HyperplaneArrangements*, for their permission to copy and modify their package’s source code.

The following is Example 6.3 of [Berkesch and Leykin 2010]:

```
i19 : R = QQ[x,y,z];
i20 : ff = toList factor ( (x^2-y^2)*(x^2-z^2)*(y^2-z^2)*z ) / first;
i21 : A = arrangement ff;
i22 : toString jumpingNumbers(A,IntervalType=>"ClosedOpen")
o22 = {{3/7, 4/7, 2/3, 6/7}, {ideal(z,y,x), ideal(z^2,y*z,x*z,y^2,x*y,x^2),
      ideal(y^2*z-z^3,x^2*z-z^3,x*y^2-x*z^2,x^2*y*y*z^2), ...
```

ACKNOWLEDGEMENTS. I am very grateful to Claudiu Raicu and Bart Snapp for their critical contributions to the package and for a number of very helpful comments about this paper and the software package itself. I would also like to thank Howard Thompson for sharing his work-in-progress and for numerous helpful conversations, Graham Denham and Gregory G. Smith, the organizers of the 2011 IMA Special Workshop on *Macaulay2*, and the organizers of the 2012 *Macaulay2* Workshop at Wake Forest.

REFERENCES.

- [Berkesch and Leykin 2010] C. Berkesch and A. Leykin, “Algorithms for Bernstein–Sato polynomials and multiplier ideals”, pp. 99–106 in *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, edited by S. M. Watt, ACM, New York, 2010. MR 2920542
- [Birkar et al. 2010] C. Birkar, P. Cascini, C. D. Hacon, and J. McKernan, “Existence of minimal models for varieties of log general type”, *J. Amer. Math. Soc.* **23**:2 (2010), 405–468. MR 2011f:14023 Zbl 1210.14019
- [Blickle 2004] M. Blickle, “Multiplier ideals and modules on toric varieties”, *Math. Z.* **248**:1 (2004), 113–121. MR 2006a:14082 Zbl 1061.14055
- [Bruns and Ichim 2010] W. Bruns and B. Ichim, “Normaliz: algorithms for affine monoids and rational cones”, *J. Algebra* **324**:5 (2010), 1098–1113. MR 2011m:20001 Zbl 1203.13033
- [Bruns and Kämpf 2010] W. Bruns and G. Kämpf, “A Macaulay2 interface for Normaliz”, *J. Softw. Algebra Geom.* **2** (2010), 15–19. MR 2881130
- [Bruns and Vetter 1988] W. Bruns and U. Vetter, *Determinantal rings*, Lecture Notes in Mathematics **1327**, Springer, Berlin, 1988. MR 89i:13001 Zbl 0673.13006

³These papers state the result for \mathbb{C} , but it holds over any field.

- [Denham and Smith 2011] G. Denham and G. G. Smith, “HyperplaneArrangements, version 0.9 — a package for Macaulay2”, 2011, <http://www.math.uiuc.edu/Macaulay2/doc/Macaulay2-1.6/share/doc/Macaulay2/HyperplaneArrangements/html/>.
- [Drton et al. 2009] M. Drton, B. Sturmfels, and S. Sullivant, *Lectures on algebraic statistics*, Oberwolfach Seminars **39**, Birkhäuser, Basel, 2009. MR 2012d:62004 Zbl 1166.13001
- [Ein and Mustață 2006] L. Ein and M. Mustață, “Invariants of singularities of pairs”, pp. 583–602 in *International Congress of Mathematicians, II*, edited by M. Sanz-Solé et al., Eur. Math. Soc., Zürich, 2006. MR 2007m:14050 Zbl 1096.14030
- [Ein et al. 2001] L. Ein, R. Lazarsfeld, and K. E. Smith, “Uniform bounds and symbolic powers on smooth varieties”, *Invent. Math.* **144**:2 (2001), 241–252. MR 2002b:13001 Zbl 1076.13501
- [Frühbis-Krüger 2014] A. Frühbis-Krüger, “Desingularization in computational applications and experiments”, pp. 269–284 in *The resolution of singular algebraic varieties* (Obergurgl, Austria, 2012.), edited by D. Ellwood et al., Amer. Math. Soc. and Clay Mathematics Institute, Providence, RI and Cambridge, MA, 2014. Zbl 1306.14001
- [González Pérez and Teissier 2002] P. D. González Pérez and B. Teissier, “Embedded resolutions of non necessarily normal affine toric varieties”, *C. R. Math. Acad. Sci. Paris* **334**:5 (2002), 379–382. MR 2003b:14019 Zbl 1052.14062
- [Hacon and McKernan 2007] C. D. Hacon and J. McKernan, “Extension theorems and the existence of flips”, pp. 76–110 in *Flips for 3-folds and 4-folds*, edited by A. Corti, Oxford Lecture Ser. Math. Appl. **35**, Oxford Univ. Press, 2007. MR 2359343 Zbl 1286.14026
- [Howald 2001] J. A. Howald, “Multiplier ideals of monomial ideals”, *Trans. Amer. Math. Soc.* **353**:7 (2001), 2665–2671. MR 2002b:14061 Zbl 0979.13026
- [Johnson 2003] A. A. Johnson, *Multiplier ideals of determinantal ideals*, Ph.D. thesis, University of Michigan, 2003, <http://search.proquest.com/docview/287933180>. MR 2704808
- [Lazarsfeld 2004] R. Lazarsfeld, *Positivity in algebraic geometry, II: Positivity for vector bundles, and multiplier ideals*, *Ergebnisse der Mathematik und ihrer Grenzgebiete (3)* **49**, Springer, Berlin, 2004. MR 2005k:14001b Zbl 1093.14500
- [Lee 2008] Y. Lee, “Chow stability criterion in terms of log canonical threshold”, *J. Korean Math. Soc.* **45**:2 (2008), 467–477. MR 2009b:14091 Zbl 1144.14300
- [Mustață 2006] M. Mustață, “Multiplier ideals of hyperplane arrangements”, *Trans. Amer. Math. Soc.* **358**:11 (2006), 5015–5023. MR 2007d:14007 Zbl 1126.14003
- [Niu 2014] W. Niu, “Singularities of generic linkage of algebraic varieties”, *Amer. J. Math.* **136**:6 (2014), 1665–1691. MR 3282984 Zbl 06380306
- [Shibuta 2011] T. Shibuta, “Algorithms for computing multiplier ideals”, *J. Pure Appl. Algebra* **215**:12 (2011), 2829–2842. MR 2012f:14037 Zbl 1233.14015
- [Siu 1998] Y.-T. Siu, “Invariance of plurigeners”, *Invent. Math.* **134**:3 (1998), 661–673. MR 99i:32035 Zbl 0955.32017
- [Teitler 2008] Z. Teitler, “A note on Mustață’s computation of multiplier ideals of hyperplane arrangements”, *Proc. Amer. Math. Soc.* **136**:5 (2008), 1575–1579. MR 2008k:14005 Zbl 1157.14001
- [Thompson 2014] H. M. Thompson, “Multiplier ideals of monomial space curves”, *Proc. Amer. Math. Soc. Ser. B* **1** (2014), 33–41. MR 3168880
- [Watanabe 2009] S. Watanabe, *Algebraic geometry and statistical learning theory*, Cambridge Monographs on Applied and Computational Mathematics **25**, Cambridge University Press, New York, 2009. MR 2011g:62185 Zbl 1180.93108

[Zwiernik 2011] P. Zwiernik, “An asymptotic behaviour of the marginal likelihood for general Markov models”, *J. Mach. Learn. Res.* **12** (2011), 3283–3310. MR 2012m:62248 Zbl 1280.68225

RECEIVED: 19 Feb 2014

REVISED: 24 Apr 2015

ACCEPTED: 5 Jun 2015

ZACH TEITLER:

zteitler@boisestate.edu

Department of Mathematics, Boise State University, 1910 University Drive, Boise, ID 83725-1555,
United States

Partially ordered sets in Macaulay2

DAVID COOK II, SONJA MAPES AND GWYNETH WHIELDON

ABSTRACT: We introduce the package *Posets* for Macaulay2. This package provides a data structure and the necessary methods for working with partially ordered sets, also called posets. In particular, the package implements methods to enumerate many commonly studied classes of posets, perform operations on posets, and calculate various invariants associated to posets.

INTRODUCTION. A *partial order* is a binary relation \leq over a set P that is antisymmetric, reflexive, and transitive. A set P together with a partial order \leq is called a *poset*, or *partially ordered set*. We refer the reader to the seminal text [Stanley 2012] for definitions omitted herein.

Posets are combinatorial structures that are used in modern mathematical research, particularly in algebra. We introduce the package *Posets* for Macaulay2 [Grayson and Stillman] via three distinct posets or related ideals which arise naturally in combinatorial algebra.

We first describe two posets that are generated from algebraic objects. The intersection semilattice associated to a hyperplane arrangement can be used to compute the number of unbounded and bounded real regions cut out by a hyperplane arrangement, as well as the dimensions of the homologies of the complex complement of a hyperplane arrangement.

Given a monomial ideal, the lcm-lattice of its minimal generators gives information on the structure of the free resolution of the original ideal. Specifically, two monomial ideals with isomorphic lcm-lattices have the “same” (up to relabeling) minimal free resolution, and the lcm-lattice can be used to compute, among other things, the multigraded Betti numbers

$$\beta_{i,b}(R/M) = \dim_K \operatorname{Tor}_{i,b}(R/M, K)$$

of the monomial ideal.

In contrast to the first two examples (associating a poset to an algebraic object), we then describe an ideal that is generated from a poset. In particular, the Hibi ideal of a finite poset is a squarefree monomial ideal which has many nice *algebraic*

MSC2010: 06A11, 13P99.

Keywords: posets.

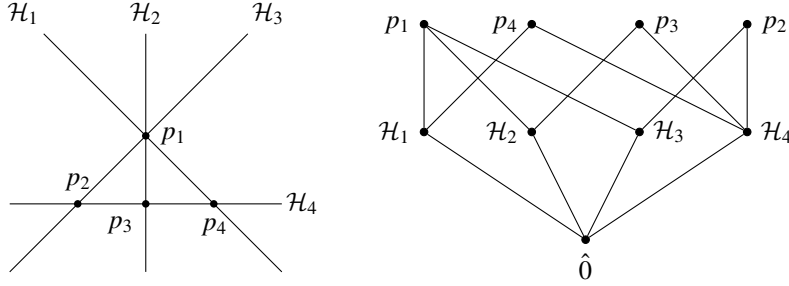


Figure 1. Left: the noncentral hyperplane arrangement \mathcal{A} comprising $\mathcal{H}_1 = V(x+y)$, $\mathcal{H}_2 = V(x)$, $\mathcal{H}_3 = V(x-y)$, $\mathcal{H}_4 = V(y+1)$. Right: The intersection meet semilattice $\mathcal{L}(\mathcal{A})$ of \mathcal{A} .

properties that can be described in terms of *combinatorial* properties of the poset. For example, the minimal free resolution, the Betti numbers, and the projective dimension are nicely described in terms of data about the poset itself.

INTERSECTION (MEET SEMI)LATTICES. A *hyperplane arrangement* \mathcal{A} is a finite collection of affine hyperplanes in some vector space V . The *dimension* of a hyperplane arrangement is defined by $\dim(\mathcal{A}) = \dim(V)$, and the *rank* of a hyperplane arrangement $\text{rank}(\mathcal{A})$ is the dimension of the span in V of the set of normals to the hyperplanes in \mathcal{A} .

The *intersection (meet semi)lattice* $\mathcal{L}(\mathcal{A})$ of \mathcal{A} is the set of the nonempty intersections of subsets of hyperplanes $\bigcap_{\mathcal{H} \in \mathcal{A}'} \mathcal{H}$ for $\mathcal{H} \in \mathcal{A}' \subseteq \mathcal{A}$, ordered by reverse inclusion. We include the empty intersection corresponding to $\mathcal{A}' = \emptyset$, which is the minimal element $\hat{0} = V$ in the intersection meet semilattice $\mathcal{L}(\mathcal{A})$. If the intersection of all hyperplanes in \mathcal{A} is nonempty, i.e., $\bigcap_{\mathcal{H} \in \mathcal{A}} \mathcal{H} \neq \emptyset$, then the intersection meet semilattice $\mathcal{L}(\mathcal{A})$ is actually a lattice. Arrangements with this property are called *central arrangements*.

Consider the noncentral hyperplane arrangement

$$\mathcal{A} = \{\mathcal{H}_1 = V(x+y), \mathcal{H}_2 = V(x), \mathcal{H}_3 = V(x-y), \mathcal{H}_4 = V(y+1)\},$$

where $\mathcal{H}_i = V(\ell_i(x, y)) \subseteq \mathbb{R}^2$ denotes the hyperplane of zeros of the linear form $\ell_i(x, y)$; see Figure 1, left. We construct $\mathcal{L}(\mathcal{A})$ in Macaulay2 as follows.

```
i1 : needsPackage "Posets";
i2 : R = RR[x,y];
i3 : A = {x + y, x, x - y, y + 1};
i4 : LA = intersectionLattice(A, R);
```

Further, using the method `texPoset` we generate \LaTeX code to display the Hasse diagram of $\mathcal{L}(\mathcal{A})$, as in Figure 1, right.

A theorem of Zaslavsky [1975] provides information about the topology of the complement of hyperplane arrangements in \mathbb{R}^n . Let μ denote the Möbius function of the intersection meet semilattice $\mathcal{L}(\mathcal{A})$. Then the number of regions that \mathcal{A} divides \mathbb{R}^n into is

$$r(\mathcal{A}) = \sum_{x \in \mathcal{L}(\mathcal{A})} |\mu(\hat{0}, x)|.$$

Moreover, the number of these regions that are bounded is

$$b(\mathcal{A}) = |\mu(\mathcal{L}(\mathcal{A}) \cup \hat{1})|,$$

where $\mathcal{L}(\mathcal{A}) \cup \hat{1}$ is the intersection meet semilattice adjoined with a maximal element.

We verify these results for the noncentral hyperplane arrangement \mathcal{A} using Macaulay2:

```
i5 : realRegions(A, R)
o5 = 10
i6 : boundedRegions(A, R)
o6 = 2
```

Moreover, in the case of hyperplane arrangements in \mathbb{C}^n , using a theorem of Orlik and Solomon [1980] we can recover the Betti numbers (dimensions of homologies) of the complement $\mathcal{M}_{\mathcal{A}} = \mathbb{C}^n - \bigcup \mathcal{A}$ of the hyperplane arrangement using purely combinatorial data of the intersection meet semilattice. In particular, $\mathcal{M}_{\mathcal{A}}$ has torsion-free integral homology with Betti numbers given by

$$\beta_i(\mathcal{M}_{\mathcal{A}}) = \dim_{\mathbb{C}}(H_i(\mathcal{M}_{\mathcal{A}})) = \sum_{\substack{x \in \mathcal{L}(\mathcal{A}) \\ \dim_{\mathbb{C}}(x) = n-i}} |\mu(\hat{0}, x)|,$$

where $\mu(\cdot)$ again represents the Möbius function. See [Wachs 2007] for details and generalizations of this formula.

Posets will compute the ranks of elements in a poset, where the ranks in the intersection meet semilattice $\mathcal{L}(\mathcal{A})$ are determined by the codimension of elements. Combining the outputs of our rank function with the Möbius function allows us to calculate that $\beta_0(\mathcal{M}_{\mathcal{A}}) = 1$, $\beta_1(\mathcal{M}_{\mathcal{A}}) = 4$, and $\beta_2(\mathcal{M}_{\mathcal{A}}) = 5$:

```
i7 : RLA = rank LA
o7 = {{ideal 0}, {ideal(x+y), ideal(x), ideal(x-y), ideal(y+1)},
      {ideal(y,x), ideal(y+1,x-1), ideal(y+1,x), ideal(y+1,x+1)}}
i8 : MF = moebiusFunction LA;
i9 : apply(RLA, r -> sum(r, x -> abs MF#(ideal 0_R, x)))
o9 = {1, 4, 5}
```

LCM-LATTICES. Let $R = K[x_1, \dots, x_t]$ be the polynomial ring in t variables over the field K , where the degree of x_i is the standard basis vector $e_i \in \mathbb{Z}^t$. Let $M = (m_1, \dots, m_n)$ be a monomial ideal in R ; then we define the *lcm-lattice* of M ,

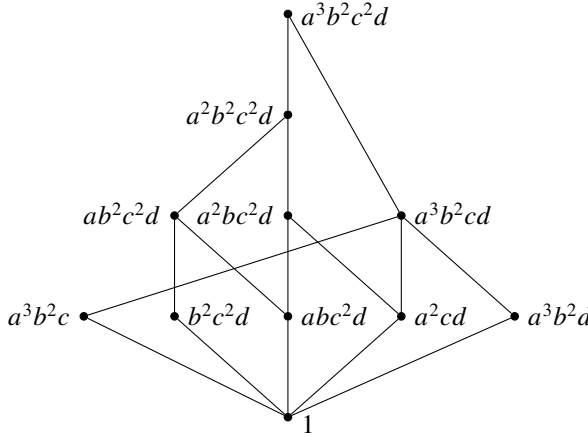


Figure 2. The lcm-lattice for $M = (a^3b^2c, a^3b^2d, a^2cd, abc^2d, b^2c^2d)$.

denoted by L_M , to be the set of all least common multiples of subsets of the generators of M , partially ordered by divisibility. It is easy to see that L_M will always be a finite atomic lattice. While lcm-lattices are nicely structured, they can be difficult to compute by hand, especially for large examples or for ideals where L_M is not ranked.

Consider the ideal

$$M = (a^3b^2c, a^3b^2d, a^2cd, abc^2d, b^2c^2d)$$

in $R = K[a, b, c, d]$. We construct L_M in Macaulay2 as follows. See Figure 2 for the Hasse diagram of L_M , as generated by the `texPoset` method.

```
i10 : R = QQ[a,b,c,d];
i11 : M = ideal(a^3*b^2*c, a^3*b^2*d, a^2*c*d, a*b*c^2*d, b^2*c^2*d);
i12 : LM = lcmLattice M;
```

Lcm-lattices, which were introduced by Gasharov, Peeva, and Welker [Gasharov et al. 1999], have become an important tool used in studying free resolutions of monomial ideals. There have been a number of results that use the lcm-lattice to give constructive methods for finding free resolutions for monomial ideals; for some examples see [Clark 2010; Peeva and Velasco 2011; Velasco 2008].

In particular, Gasharov, Peeva, and Welker [Gasharov et al. 1999] provided a key connection between the lcm-lattice of a monomial ideal M of R and its minimal free resolution; namely, one can compute the (multigraded) Betti numbers of R/M using the lcm-lattice. Let $\Delta(P)$ denote the order complex of the poset P ; then for $i \geq 1$ we have

$$\beta_{i,b}(R/M) = \dim \tilde{H}_{i-2}(\Delta(\hat{0}, b); K)$$

for all $b \in L_M$, and so

$$\beta_i(R/M) = \sum_{b \in L_M} \dim \tilde{H}_{i-2}(\Delta(\hat{0}, b); K).$$

These computations can all be done using *Posets* together with the package *SimplicialComplexes* by S. Popescu, G. Smith, and M. Stillman. In particular, we show that $\beta_{i,a^2b^2c^2d} = 0$ for all i with the following calculation:

```
i13 : D1 = orderComplex(openInterval(LM, 1_R, a^2*b^2*c^2*d));
i14 : prune HH(D1)
o14 = -1 : 0
      0 : 0
      1 : 0
o14 : GradedModule
```

Similarly, we show that $\beta_{1,a^3b^2cd} = 2$:

```
i15 : D2 = orderComplex(openInterval(LM, 1_R, a^3*b^2*c*d));
i16 : prune HH(D2)
o16 = -1 : 0
      2
      0 : QQ
o16 : GradedModule
```

HIBI IDEALS. Let $P = \{p_1, \dots, p_n\}$ be a finite poset with partial order \leq , and let K be a field. The *Hibi ideal*, introduced by Herzog and Hibi [2005], of P over K is the squarefree ideal H_P in $R = K[x_1, \dots, x_n, y_1, \dots, y_n]$ generated by the monomials

$$u_I := \prod_{p_i \in I} x_i \prod_{p_i \notin I} y_i,$$

where I is an order ideal of P , i.e., for every $i \in I$ and $p \in P$, if $p \leq i$ then $p \in I$.

NB: The Hibi ideal is the ideal of the monomial generators of the Hibi ring, a toric ring first described by Hibi [1987].

```
i17 : P = divisorPoset 12;
i18 : HP = hibiIdeal P
o18 = monomialIdeal (x x x x x x , x x x x x y , x x x x y y ,
                    0 1 2 3 4 5   0 1 2 3 4 5   0 1 2 4 3 5 ,
                    x x x x y y y , x x x y y y y , x x x y y y y , x x y y y y y ,
                    0 1 2 3 4 5   0 1 3 2 4 5   0 1 2 3 4 5   0 2 1 3 4 5 ,
                    x x y y y y y , x y y y y y y , y y y y y y y )
                    0 1 2 3 4 5   0 1 2 3 4 5   0 1 2 3 4 5
```

Herzog and Hibi [2005] proved that every power of H_P has a linear resolution, and the i -th Betti number $\beta_i(R/H_P)$ is the number of intervals of the distributive lattice $\mathcal{L}(P)$ of P isomorphic to the rank- i boolean lattice. Using [Stanley 2012, Exercise 3.47], we recover this by looking instead at the number of elements of $\mathcal{L}(P)$ that cover exactly i elements:


```

i19 : betti res HP
      0  1  2  3
o19 = total: 1 10 12 3
      0: 1  .  .  .
      1: .  .  .  .
      2: .  .  .  .
      3: .  .  .  .
      4: .  .  .  .
      5: . 10 12 3

i20 : LP = distributiveLattice P;
i21 : cvrs = partition(last, coveringRelations LP);
i22 : -- Determine the number of elements each element covers.
      iCvrs = tally apply(keys cvrs, i -> #cvrs#i);
i23 : -- Turn iCvrs into a list indexed by integers.
      gk = prepend(1, apply(sort keys iCvrs, k -> iCvrs#k))
o23 : {1, 6, 3}
i24 : -- Determine the number of intervals of LP isomorphic
      -- to boolean lattices of a given rank.
      apply(#gk, i -> sum(i..<#gk, j -> binomial(j, i) * gk_j))
o24 : {10, 12, 3}

```

Moreover, Herzog and Hibi [2005] proved that the projective dimension of H_P is the Dilworth number of P , i.e., the maximum length of an antichain of P .

```

i25 : pdim module HP == dilworthNumber P
o25 = true

```

REFERENCES.

- [Clark 2010] T. B. P. Clark, “Poset resolutions and lattice-linear monomial ideals”, *J. Algebra* **323**:4 (2010), 899–919. MR 2011b:13037 Zbl 1203.13016
- [Gasharov et al. 1999] V. Gasharov, I. Peeva, and V. Welker, “The lcm-lattice in monomial resolutions”, *Math. Res. Lett.* **6**:5-6 (1999), 521–532. MR 2001e:13018 Zbl 0970.13004
- [Grayson and Stillman] D. R. Grayson and M. E. Stillman, “Macaulay2: a software system for research in algebraic geometry”, Available at <http://www.math.uiuc.edu/Macaulay2>.
- [Herzog and Hibi 2005] J. Herzog and T. Hibi, “Distributive lattices, bipartite graphs and Alexander duality”, *J. Algebraic Combin.* **22**:3 (2005), 289–302. MR 2006h:06004 Zbl 1090.13017
- [Hibi 1987] T. Hibi, “Distributive lattices, affine semigroup rings and algebras with straightening laws”, pp. 93–109 in *Commutative algebra and combinatorics* (Kyoto, 1985), edited by M. Nagata and H. Matsumura, Adv. Stud. Pure Math. **11**, North-Holland, Amsterdam, 1987. MR 90b:13024 Zbl 0654.13015
- [Orlik and Solomon 1980] P. Orlik and L. Solomon, “Combinatorics and topology of complements of hyperplanes”, *Inventiones Math.* **56**:2 (1980), 167–189. MR 81e:32015 Zbl 0432.14016
- [Peeva and Velasco 2011] I. Peeva and M. Velasco, “Frames and degenerations of monomial resolutions”, *Trans. Amer. Math. Soc.* **363**:4 (2011), 2029–2046. MR 2011k:13021 Zbl 1221.13024
- [Stanley 2012] R. P. Stanley, *Enumerative combinatorics, I*, 2nd ed., Cambridge Studies in Advanced Mathematics **49**, Cambridge University Press, 2012. MR 2868112 Zbl 1247.05003
- [Velasco 2008] M. Velasco, “Minimal free resolutions that are not supported by a CW-complex”, *J. Algebra* **319**:1 (2008), 102–114. MR 2008j:13028 Zbl 1133.13015

- [Wachs 2007] M. L. Wachs, “Poset topology: tools and applications”, pp. 497–615 in *Geometric combinatorics*, edited by E. Miller et al., IAS/Park City Math. Ser. **13**, Amer. Math. Soc., Providence, RI, 2007. MR 2383132 Zbl 1135.06001
- [Zaslavsky 1975] T. Zaslavsky, *Facing up to arrangements: face-count formulas for partitions of space by hyperplanes*, Mem. Amer. Math. Soc. **154**, Amer. Math. Soc., Providence, RI, 1975. MR 50 #9603 Zbl 0296.50010

RECEIVED: 21 Jan 2014

REVISED: 14 Aug 2014

ACCEPTED: 5 Jun 2015

DAVID COOK II:

dwcook@eiu.edu

Department of Mathematics & Computer Science, Eastern Illinois University, Charleston, IN 46613, United States

SONJA MAPES:

smapes1@nd.edu

Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556, United States

GWYNETH WHIELDON:

whieldon@hood.edu

Department of Mathematics, Hood College, Frederick, MD 21701, United States

Free resolutions and modules with a semisimple Lie group action

FEDERICO GALETTO

ABSTRACT: We introduce the package *HighestWeights* for Macaulay2. This package provides tools to study the representation-theoretic structure of free resolutions and graded modules over a polynomial ring with the action of a semisimple Lie group. The methods of this package allow users to consider the free modules in a resolution, or the graded components of a module, as representations of a semisimple Lie group by means of their weights, and to obtain their decomposition into highest-weight representations.

1. INTRODUCTION. Let R be a polynomial ring over the complex numbers with a \mathbb{Z}^m -grading, and let M be a finitely generated graded R -module. Under mild assumptions on R , for every degree $d \in \mathbb{Z}^m$ each graded component M_d is a finite-dimensional complex vector space. Next assume that G is a complex Lie group and that there is a degree-preserving \mathbb{C} -linear action of G on R compatible with multiplication, i.e., such that for every $g \in G$, $r_1, r_2 \in R$ we have $g \cdot (r_1 r_2) = (g \cdot r_1)(g \cdot r_2)$. We are interested in those R -modules M with a degree-preserving \mathbb{C} -linear action of G compatible with the module structure, i.e., such that for every $g \in G$, $r \in R$ and $m \in M$ we have $g \cdot (rm) = (g \cdot r)(g \cdot m)$. Notice that each graded component of such a module M is stable under the action of G .

Examples of such modules can arise naturally. For instance, let X be a finite-dimensional representation of a complex Lie group G . The symmetric algebra $\text{Sym}(X)$, with the standard grading determined by $\text{Sym}^1(X) \cong X$, is an example of a polynomial ring with a degree-preserving \mathbb{C} -linear action of G . The action of G also extends to the projective space $\mathbb{P}(X)$. If V is a projective variety in $\mathbb{P}(X)$ which is fixed by the action of G , then the affine cone \hat{V} is an affine variety in X which is fixed by G . Moreover, \hat{V} is the zero locus of some radical homogeneous ideal I in $\text{Sym}(X)$, and the quotient ring $\text{Sym}(X)/I$, i.e., the affine coordinate ring of \hat{V} , is an example of a $\text{Sym}(X)$ -module with a compatible G -action.

MSC2010: primary 13P20; secondary 22E46.

Keywords: equivariant free resolution, irreducible representation, highest weight, algebraic torus, semisimple Lie group, decomposition algorithm.

Let M be a finitely generated graded R -module with the kind of G -action described earlier. Denote by \mathfrak{m} the maximal ideal generated by the variables in R , and set $V_0 = M/\mathfrak{m}M$. The vector space V_0 is a finite-dimensional graded representation of G . Assuming G is linearly reductive, the natural projection $M \rightarrow V_0$ admits a section $V_0 \rightarrow M$ which is compatible with the grading and G -action. This section extends to an R -linear map $d_0 : F_0 \rightarrow M$, where F_0 is defined to be $V_0 \otimes_{\mathbb{C}} R$. By construction, F_0 is a graded free R -module with a natural action of G that commutes with d_0 ; moreover, d_0 maps a basis of F_0 to a minimal generating set of M . Notice also that $F_0/\mathfrak{m}F_0 \cong V_0$ as a graded representation of G . Next, let $V_1 = \ker(d_0)/\mathfrak{m}\ker(d_0)$. Using the same ideas as before, let F_1 be $V_1 \otimes_{\mathbb{C}} R$ (so that $F_1/\mathfrak{m}F_1 \cong V_1$) and define the map $d_1 : F_1 \rightarrow F_0$. Again, d_1 will be a map of graded free R -modules that commutes with the action of G and maps a basis of F_1 to a minimal generating set of $\ker(d_0)$; in addition, $d_1 \circ d_0 = 0$. Iterating this procedure constructs a minimal free resolution of M with a built-in action of G ; in particular, the procedure is guaranteed to terminate. Since any two minimal free resolutions of M are isomorphic as complexes, the action of G transfers to every other minimal free resolution via the isomorphism with the one just constructed.

Summarizing what we said so far, for every finitely generated graded R -module M with a degree-preserving G -action compatible with the module structure, the following occurs:

- (1) For every degree $d \in \mathbb{Z}^m$, the graded component M_d is a finite-dimensional representation of G .
- (2) If the complex

$$0 \longleftarrow M \longleftarrow F_0 \longleftarrow F_1 \longleftarrow \cdots \longleftarrow F_n \longleftarrow 0$$

is a minimal free resolution of M as an R -module, then the action of G on M extends to each F_i and there is an isomorphism $F_i \cong (F_i/\mathfrak{m}F_i) \otimes_{\mathbb{C}} R$ of graded R -modules with a G -action. Each $F_i/\mathfrak{m}F_i$ is a finite-dimensional graded vector space so, for each degree $d \in \mathbb{Z}^m$, $(F_i/\mathfrak{m}F_i)_d$ is a finite-dimensional representation of G .

When the group G is semisimple, it is a typical problem to decompose a finite-dimensional representation into irreducible representations. Moreover each irreducible representation is indexed by a so-called highest weight. The main purpose of *HighestWeights* is to provide users of Macaulay2 [Grayson and Stillman] with tools to obtain the highest-weight decomposition of the representations M_d and $(F_i/\mathfrak{m}F_i)_d$ introduced above. This purpose is achieved by implementing an algorithm for propagating weights of tori along equivariant maps introduced in [Galetto 2015].

This article is organized as follows: the next section details the mathematical assumptions for using this package, Section 3 presents two examples in detail, and Section 4 contains some final remarks. We have also included a brief review of the representation theory of semisimple Lie groups in the Appendix.

2. PACKAGE ASSUMPTIONS. Before presenting some examples, we discuss some assumptions of this package. Using the notation of the introduction, the polynomial ring R must be positive \mathbb{Z}^m -graded for some positive integer m , in the sense of [Kreuzer and Robbiano 2005, Definition 4.2.4]. More explicitly, if $R = \mathbb{C}[x_1, \dots, x_n]$, then R is graded by elements of \mathbb{Z}^m in such a way that:

- (1) each nonzero constant in R has degree $0 \in \mathbb{Z}^m$;
- (2) the degree of each variable $x_i \in R$ is a nonzero vector in \mathbb{Z}^m , and its first nonzero entry is positive;
- (3) the matrix with rows given by the degrees of the variables x_1, \dots, x_n has rank m .

This ensures that, for every finitely generated graded R -module M and for every degree $d \in \mathbb{Z}^m$, each graded component M_d is a finite-dimensional complex vector space. The variables in the polynomial ring R must be weight vectors for the action of the chosen maximal torus in G ; this can always be achieved up to a linear change of variables in R . The user is expected to provide the weight of each variable. Any monomial ordering on the monomials of R is allowed. All free R -modules must be endowed with a term over position up/down or position up/down over term ordering; in Macaulay2 this is established with the declaration of the ring. The default ordering, term over position up, is fine for most computations, unless the user needs a different one.

To obtain the decomposition of a graded component M_d , the user is expected to provide a presentation $\varphi : F_1 \rightarrow F_0$ of M in the form of a matrix written with respect to a homogeneous basis $\{e_1, \dots, e_r\}$ of F_0 such that the residue classes $\bar{e}_1, \dots, \bar{e}_r$ modulo $\mathfrak{m}F_0$ form a basis of weight vectors of $F_0/\mathfrak{m}F_0$; the user will also need to provide a list with the weights of $\bar{e}_1, \dots, \bar{e}_r$. For modules with a compatible group action, presentations of this kind are, in our experience, the most natural. As for resolutions, the user must provide a list of weights for a basis e_1, \dots, e_r as before, for any one of the modules F_i . When $M = R/I$, for a G -stable ideal I in R , the module F_0 is simply R with a trivial G -action; in this case, the user does not need to input any weight (other than those of the variables of R).

The package *WeylGroups*, which is loaded automatically by this package, is used to declare the type of a semisimple group and to handle many weight-related operations behind the scenes. However, for the purpose of this package, weights

are to be provided simply as lists of integers, not as objects of type `Weight` as in `WeylGroups`.

3. EXAMPLES. For an explanation of the notations and conventions relating to weights that appear in the following examples, we refer the reader to Section A2.

3.1. The coordinate ring of the Grassmannian. Let $E = \mathbb{C}^6$, the standard representation of $\mathrm{SL}_6(\mathbb{C})$, with coordinate basis $\{e_0, \dots, e_5\}$. The Grassmannian $V = \mathrm{Gr}(2, E^*)$ is the projective variety which parametrizes two-dimensional subspaces of E^* ; it is embedded in $\mathbb{P}(\wedge^2 E^*)$ using the Plücker equations [Shafarevich 1994, Chapter I, §4.1]. Consider $\wedge^2 E^*$ as a complex affine space. Let C be the affine cone over V , i.e., the subvariety of $\wedge^2 E^*$ which is the union of all the one-dimensional subspaces of $\wedge^2 E^*$ belonging to V . The space $\wedge^2 E^*$ has a natural action of $\mathrm{SL}_6(\mathbb{C})$ which fixes C .

Our polynomial ring R is the ring of polynomial functions over $\wedge^2 E^*$, i.e., the symmetric algebra $\mathrm{Sym}(\wedge^2 E)$. The elements $p_{i,j} = e_i \wedge e_j$ for $0 \leq i < j \leq 5$ form a basis of weight vectors of $\wedge^2 E$ and will be the variables in R . The defining ideal of C is generated by the Plücker equations; this ideal, which we call I , can be conveniently obtained in Macaulay2 using the command `Grassmannian`. We resolve the quotient R modulo I as an R -module and call `RI` the minimal free resolution:

```
i1 : printWidth=72; truncateOutput 200;
i3 : I=Grassmannian(1,5,CoefficientRing=>QQ); R=ring I;
o3 : Ideal of QQ[p0,1, p0,2, p1,2, p0,3, p1,3, p2,3, p0,4, p1,4, p2,4, ... ]
i5 : RI=res I; betti RI
      0  1  2  3  4  5  6
o6 = total: 1 15 35 42 35 15 1
      0: 1 . . . . .
      1: . 15 35 21 . .
      2: . . . 21 35 15 .
      3: . . . . . 1
o6 : BettiTally
```

Now we load the package and assign weights to the variables of R . First we input the weights of e_0, \dots, e_5 in a list `L`.

```
i7 : loadPackage "HighestWeights";
i8 : L={{1,0,0,0,0},{-1,1,0,0,0},{0,-1,1,0,0},{0,0,-1,1,0},{0,0,0,-1,1},
      {0,0,0,0,-1}};
```

The weight of $p_{i,j} = e_i \wedge e_j$ is equal to the sum of the weights of e_i and e_j (see Appendix A2). The subscripts of the variables $p_{i,j}$ are the elements of subsets $(\{0, 1, 2, 3, 4, 5\}, 2)$, the 2-subsets of the set $\{0, 1, 2, 3, 4, 5\}$. Hence

taking sums of pairs of weights in L over this indexing set will give us a complete list of weights for the variables $p_{i,j}$, as listed by Macaulay2:

```
i9 : W=apply(subsets({0,1,2,3,4,5},2),s->L_(s_0)+L_(s_1))
o9 = {{0, 1, 0, 0, 0}, {1, -1, 1, 0, 0}, {-1, 0, 1, 0, 0}, {1, 0, -1, 1, 0},
-----
      {-1, 1, -1, 1, 0}, {0, -1, 0, 1, 0}, {1, 0, 0, -1, 1}, {-1, 1, 0, -1, 1},
-----
      {0, -1, 1, -1, 1}, {0, 0, -1, 0, 1}, {1, 0, 0, 0, 0}, ...
o9 : List
```

We declare D to be the Dynkin type A_5 , which is the type of the group $SL_6(\mathbb{C})$. We then attach the weights in W to the variables in R with the command `setWeights`; the arguments are the ring, the type and the weights of the variables, respectively. The output will be the highest-weight decomposition of the \mathbb{C} -linear subspace of R generated by its variables; it is given in the form of a Tally, with keys describing the highest weights of the irreducible representation appearing in the decomposition and values equal to the multiplicities of those representations. In this case, we get simply $\{0, 1, 0, 0, 0\} \Rightarrow 1$, which means that the decomposition contains only one copy of the irreducible representation with highest weight $\{0, 1, 0, 0, 0\}$, i.e., $\bigwedge^2 E$, as expected:

```
i10 : D=dynkinType{"A",5}; setWeights(R,D,W)
o11 = Tally{{0, 1, 0, 0, 0} => 1}
o11 : Tally
```

All monomials in R are weight vectors. To recover the weight of a monomial, use the command `getWeights` with the monomial as the argument:

```
i12 : getWeights(p_(0,1)*p_(1,2))
o12 = {-1, 1, 1, 0, 0}
o12 : List
```

We can now issue the command `highestWeightsDecomposition` to obtain the decomposition of the representations corresponding to the free modules in the resolution; the only argument is the resolution RI . Suppose the free modules in RI are F_0, \dots, F_6 . The outermost HashTable in the output has keys equal to the subscripts of the free modules in RI . The value corresponding to a key i is itself a HashTable with keys equal to the degrees of the generators of F_i . Finally the value corresponding to a certain degree d is a Tally containing the highest-weight decomposition of the representation $(F_i/mF_i)_d$, as described earlier:

```
i13 : highestWeightsDecomposition(RI)
o13 = HashTable{0 => HashTable{{0} => Tally{{0, 0, 0, 0, 0} => 1}}}
      1 => HashTable{{2} => Tally{{0, 0, 0, 1, 0} => 1}}
      2 => HashTable{{3} => Tally{{1, 0, 0, 0, 1} => 1}}
```



```

3 => HashTable{{4} => Tally{{2, 0, 0, 0, 0} => 1}}
      {5} => Tally{{0, 0, 0, 0, 2} => 1}}
4 => HashTable{{6} => Tally{{1, 0, 0, 0, 1} => 1}}
5 => HashTable{{7} => Tally{{0, 1, 0, 0, 0} => 1}}
6 => HashTable{{9} => Tally{{0, 0, 0, 0, 0} => 1}}

```

```
o13 : HashTable
```

By analyzing this output, we obtain the following description for RI :

$$\begin{aligned}
R \leftarrow \bigwedge^4 E \otimes R(-2) &\leftarrow \mathbb{S}_{2,1,1,1,1} E \otimes R(-3) \\
&\leftarrow \mathbb{S}_2 E \otimes R(-4) \oplus \mathbb{S}_{2,2,2,2,2} E \otimes R(-5) \leftarrow \mathbb{S}_{2,1,1,1,1} E \otimes R(-6) \\
&\leftarrow \bigwedge^2 E \otimes R(-7) \leftarrow R(-9) \leftarrow 0.
\end{aligned}$$

Next we turn to the coordinate ring of C , i.e., the quotient ring $Q = R/I$. We decompose its graded components in the range of degrees from 0 to 4, again with the command `highestWeightsDecomposition`. This time the arguments are the ring followed by the lowest and highest degrees in the range to be decomposed:

```

i14 : Q=R/I; highestWeightsDecomposition(Q,0,4)
o15 = HashTable{0 => Tally{{0, 0, 0, 0, 0} => 1}}
      1 => Tally{{0, 1, 0, 0, 0} => 1}}
      2 => Tally{{0, 2, 0, 0, 0} => 1}}
      3 => Tally{{0, 3, 0, 0, 0} => 1}}
      4 => Tally{{0, 4, 0, 0, 0} => 1}}
o15 : HashTable

```

We deduce that $(R/I)_d = \mathbb{S}_{d,d} E$ for $d \in \{0, \dots, 4\}$. We can also decompose the graded components of the ring R in a range of degrees or in a single degree:

```

i16 : highestWeightsDecomposition(R,2)
o16 = Tally{{0, 0, 0, 1, 0} => 1}}
      {0, 2, 0, 0, 0} => 1
o16 : Tally

```

For example, $R_2 = \bigwedge^4 E \oplus \mathbb{S}_{2,2} E$. Since the representation $\bigwedge^4 E$ appears in R_2 but not in $(R/I)_2$, we deduce that it must be in I_2 , the graded component of I of degree 2. This can be verified directly by decomposing I_2 as follows:

```

i17 : highestWeightsDecomposition(I,2)
o17 = Tally{{0, 0, 0, 1, 0} => 1}}
o17 : Tally

```

3.2. The Buchsbaum–Rim complex. Let $E = \mathbb{C}^6$ with coordinate basis $\{e_1, \dots, e_6\}$ and $F = \mathbb{C}^3$ with coordinate basis $\{f_1, f_2, f_3\}$. Denote by R the symmetric algebra $\text{Sym}(E \otimes F)$; R is a polynomial ring with variables $x_{i,j} = e_i \otimes f_j$. We take M to be the cokernel of a generic 3×6 matrix of variables in R . The minimal free resolution of M is an example of a Buchsbaum–Rim complex [Eisenbud 1995, Appendix A2.6]. We call this complex BR.

```

i1 : printWidth = 72; truncateOutput 200;
i3 : R=QQ[x_(1,1)..x_(6,3)];
i4 : G=genericMatrix(R,3,6)
o4 = | x_(1,1) x_(2,1) x_(3,1) x_(4,1) x_(5,1) x_(6,1) |
      | x_(1,2) x_(2,2) x_(3,2) x_(4,2) x_(5,2) x_(6,2) |
      | x_(1,3) x_(2,3) x_(3,3) x_(4,3) x_(5,3) x_(6,3) |
o4 : Matrix R <--- R
i5 : M=coker G; BR=res M; betti BR
o7 = total: 0 1 2 3 4
           0: 3 6 . . .
           1: . . . . .
           2: . . 15 18 6
o7 : BettiTally

```

The ring R carries a degree compatible action of $\mathrm{SL}_6(\mathbb{C}) \times \mathrm{SL}_3(\mathbb{C})$. Define the map of graded free R -modules

$$\varphi : E \otimes R(-1) \rightarrow F^* \otimes R, \quad e_i \otimes 1 \mapsto \sum_{j=1}^3 f_j^* \otimes x_{i,j},$$

where $\{f_1^*, f_2^*, f_3^*\}$ is the dual basis in F^* . The matrix of φ with respect to the bases $\{e_1 \otimes 1, \dots, e_6 \otimes 1\}$ and $\{f_1^* \otimes 1, f_2^* \otimes 1, f_3^* \otimes 1\}$ is precisely the generic matrix G introduced above. Moreover, φ is $\mathrm{SL}_6(\mathbb{C}) \times \mathrm{SL}_3(\mathbb{C})$ -equivariant, meaning that for all $g \in \mathrm{SL}_6(\mathbb{C}) \times \mathrm{SL}_3(\mathbb{C})$, $e \in E$ and $r \in R$ we have $\varphi(g \cdot (e \otimes r)) = g \cdot \varphi(e \otimes r)$. This makes its cokernel M a module with a compatible $\mathrm{SL}_6(\mathbb{C}) \times \mathrm{SL}_3(\mathbb{C})$ -action.

The weight of $x_{i,j} = e_i \otimes f_j$ is obtained by concatenating the weight of e_i with that of f_j . First we record the weights of e_1, \dots, e_6 in a list `e` and those of f_1, f_2, f_3 in a list `f`. Then we concatenate them as illustrated below and attach the resulting list to the variables $x_{i,j}$. Care must be taken that the order of the weights matches the order of the variables:

```

i8 : loadPackage "HighestWeights";
i9 : e={{1,0,0,0,0},{-1,1,0,0,0},{0,-1,1,0,0},{0,0,-1,1,0},{0,0,0,-1,1},
      {0,0,0,0,-1}};
i10 : f={{1,0},{-1,1},{0,-1}};
i11 : W=flatten table(e,f,(u,v)->u|v)
o11 = {{1, 0, 0, 0, 0, 1, 0}, {1, 0, 0, 0, 0, -1, 1}, {1, 0, 0, 0, 0, 0, -1},
      -----
      {-1, 1, 0, 0, 0, 1, 0}, {-1, 1, 0, 0, 0, -1, 1}, {-1, 1, 0, 0, 0, 0, -1},
      -----
      {0, -1, 1, 0, 0, 1, 0}, {0, -1, 1, 0, 0, -1, 1}, { ...
o11 : List
i12 : D=dynkinType{"A",5},{"A",2}}; setWeights(R,D,W)

```

```
o13 = Tally{{1, 0, 0, 0, 0, 1, 0} => 1}
o13 : Tally
```

In order to decompose the representations in a resolution, we need to ensure that the coordinate basis for at least one of the free modules in the resolution is a basis of weight vectors, and then we need to input the weights of the elements of that basis. For our resolution BR, we could choose the first or the second free module. In fact, the first differential of BR is the map $\varphi : E \otimes R(-1) \rightarrow F^* \otimes R$ whose matrix was written with respect to the bases of weight vectors $\{e_1 \otimes 1, \dots, e_6 \otimes 1\}$ and $\{f_1^* \otimes 1, f_2^* \otimes 1, f_3^* \otimes 1\}$. We choose to work with the first module, i.e., the codomain of φ . Notice that the element $1 \in R$ appearing in the tensor product has weight zero; hence it does not contribute to the weight of the basis elements. Also the $\mathrm{SL}_6(\mathbb{C})$ factor of our group acts trivially on F^* , and hence to obtain the weight of $f_1^* \otimes 1$ we concatenate $\{0, 0, 0, 0, 0\}$, the weight of the trivial representation of $\mathrm{SL}_6(\mathbb{C})$, with $\{-1, 0\}$, the weight of f_1^* . We proceed similarly for the other basis vectors and record the weights in the list U0:

```
i14 : U0={{0,0,0,0,0,-1,0},{0,0,0,0,0,1,-1},{0,0,0,0,0,0,1}};
```

At this point we are ready to decompose BR. To do so, we issue the command `highestWeightsDecomposition` with three arguments: the first is BR, the second is an integer i informing Macaulay2 that we wish to provide the weights in the i -th free module of the complex, and the third is the list of weights in the coordinate basis of the i -th module (remember the indexing of the modules starts from zero in Macaulay2):

```
i15 : H0=highestWeightsDecomposition(BR,0,U0)
o15 = HashTable{0 => HashTable{{0} => Tally{{0, 0, 0, 0, 0, 0, 1} => 1}}}
      1 => HashTable{{1} => Tally{{1, 0, 0, 0, 0, 0, 0} => 1}}
      2 => HashTable{{4} => Tally{{0, 0, 0, 1, 0, 0, 0} => 1}}
      3 => HashTable{{5} => Tally{{0, 0, 0, 0, 1, 1, 0} => 1}}
      4 => HashTable{{6} => Tally{{0, 0, 0, 0, 0, 2, 0} => 1}}
o15 : HashTable
```

We deduce that BR decomposes as

$$F^* \otimes R \leftarrow E \otimes R(-1) \leftarrow \bigwedge^4 E \otimes R(-4) \leftarrow \bigwedge^5 E \otimes F \otimes R(-5) \leftarrow \mathbb{S}_2 F \otimes R(-6) \leftarrow 0.$$

If we choose to start from the second module, we need to provide the list of weights of the elements $e_1 \otimes 1, \dots, e_6 \otimes 1$. The commands are:

```
i16 : U1={{1,0,0,0,0,0,0},{-1,1,0,0,0,0,0},{0,-1,1,0,0,0,0},{0,0,-1,1,0,0,0},
          {0,0,0,-1,1,0,0},{0,0,0,0,-1,0,0}};
i17 : H1=highestWeightsDecomposition(BR,1,U1); H0==H1
o18 = true
```

Indeed, the decomposition is the same.

As with rings and ideals, we can decompose the graded components of a module. The difference is that we need to provide a list of weights for the generators of the presentation used to define the module. For our module M , this is exactly the list $U0$ introduced earlier. As usual, we may decompose a single degree or a range.

```
i19 : highestWeightsDecomposition(M,-1,2,U0)
o19 = HashTable{-1 => Tally{}}
      0 => Tally{{0, 0, 0, 0, 0, 0, 1} => 1}
      1 => Tally{{1, 0, 0, 0, 0, 1, 1} => 1}
      2 => Tally{{0, 1, 0, 0, 0, 0, 2} => 1}
          {2, 0, 0, 0, 0, 2, 1} => 1
o19 : HashTable
```

Since M is generated in degree zero, we see that the output contains an empty decomposition in degree -1 . Whereas we see, for example, that

$$M_2 = \wedge^2 E \otimes S_{2,2} F \oplus S_2 E \otimes S_{3,1} F.$$

4. CLOSING REMARKS. Here we comment on a few points of (potential) interest.

- The method `highestWeightsDecomposition` provides the main functionality of this package. This method relies on the method `propagateWeights` and the function `decomposeWeightsList`, both of which are also exported. The method `propagateWeights` implements (with minor changes) an algorithm of [Galetto 2015]. The function `decomposeWeightsList` implements a modified version of Freudenthal's multiplicity formula using the algorithm discussed in [de Graaf 2000, Chapter 8.9] and [Moody and Patera 1982]. We do not anticipate the user employing these commands directly, but they are available for those who wish to experiment with them. More details are available in the package documentation.
- Multigradings are supported as well as single gradings, whenever they are compatible with the group action. An example is included in the documentation that involves multigradings.
- Decomposing graded components of rings and modules tends to work better in low degrees, as the dimension of graded components can grow fast.
- All the examples presented in this article and in the documentation of this package are over the field \mathbb{C} of complex numbers for representation-theoretic reasons; however, all computations are performed in Macaulay2 over the field \mathbb{Q} of rational numbers. The reader interested in an explanation of why the computational results obtained over \mathbb{Q} can be interpreted over \mathbb{C} may consult [Galetto 2015].
- For further concrete examples where the package *HighestWeights* may be useful, including links to Macaulay2 files, see [Galetto 2014]. The Macaulay2 package *PieriMaps* [Sam 2009] provides means to construct additional examples.

APPENDIX: SOME REPRESENTATION THEORY. We present here a brief review of the representation theory of semisimple Lie groups with the terminology used in this paper and by the package *HighestWeights*. For more details the reader can consult [Humphreys 1975; 1978; Fulton and Harris 1991]. The manual of the software package *LiE* [van Leeuwen et al. 1992] also contains a brief, and in our opinion well-written, review of the theory.

All Lie groups are intended to be complex Lie groups.

A1. Type and representations of a semisimple Lie group. A (simply connected) simple Lie group is one of the following: the special linear group $\mathrm{SL}_{n+1}(\mathbb{C})$ (type A_n), the spin group $\mathrm{Spin}_{2n+1}(\mathbb{C})$ (type B_n) or $\mathrm{Spin}_{2n}(\mathbb{C})$ (type D_n), the symplectic group $\mathrm{Sp}_{2n}(\mathbb{C})$ (type C_n), or one of the exceptional groups of type E_6 , E_7 , E_8 , F_4 and G_2 . Every semisimple Lie group G is the quotient of a finite product of the simple Lie groups above, called the simple components of G , by a finite subgroup. The type of G is then obtained by concatenating the types of the simple components.

Let \mathbb{C}^\times be the multiplicative group of nonzero complex numbers. A *torus* is a Lie group which is isomorphic to $(\mathbb{C}^\times)^n$, for some positive integer n called the rank of the torus. Every semisimple Lie group G contains a maximal torus. All maximal tori are conjugate and hence have the same rank; the rank of G is defined to be the rank of a maximal torus. Usually one particular maximal torus is fixed, and it is denoted by T . The character group of T , denoted by $X(T)$, is the set of all Lie group homomorphisms $\chi : T \rightarrow \mathbb{C}^\times$. For any finite-dimensional representation V of G , there is a unique decomposition $V = \bigoplus_{\chi \in X(T)} V_\chi$, where $V_\chi = \{v \in V \mid \forall t \in T, t \cdot v = \chi(t)v\}$. The characters χ such that $V_\chi \neq 0$ are called *weights* of V , and $\dim V_\chi$ is called the multiplicity of χ in V . Each subspace V_χ is called a weight space, and its nonzero elements are called weight vectors with weight χ . The weights of V along with their multiplicities uniquely determine V as a representation of G . Moreover, if T has rank n , there is a group isomorphism $X(T) \cong \mathbb{Z}^n$, and hence weights can be simply recorded as lists of integers.

Every semisimple Lie group G contains a maximal connected solvable subgroup B , called Borel subgroup, which contains the fixed maximal torus T . Let V be a finite-dimensional representation of G and let $v \in V$ be a weight vector of weight ω . If v spans a B -stable one-dimensional subspace of V , then v is said to be a *highest-weight vector*. The representation V is irreducible if and only if v is, up to multiplication by a scalar, the only highest-weight vector in V . In this case, V is the unique irreducible representation of G with highest weight ω , and it is often denoted by $V(\omega)$.

A simple Lie group S of rank n has n fundamental representations. The construction of fundamental representations for each type is detailed in [Fulton and

Harris 1991], and a summary can be found in [Tits 1967]. The concept of fundamental representations can be extended to any semisimple Lie group G : if G has simple components S_1, \dots, S_r , then for each component S_i and for each fundamental representation $V_{i,j}$ of S_i , take $V_{i,j}$ to be a fundamental representation of G with a trivial action of all the other components. Fundamental representations are irreducible, and their highest weights are called *fundamental weights*. The fundamental weights are often denoted $\omega_1, \dots, \omega_n$; they form a basis of $X(T)$ as a free abelian group, and the isomorphism $X(T) \cong \mathbb{Z}^n$ is usually taken to send them to the coordinate basis of \mathbb{Z}^n . From now on we will always write weights as elements of the additive group \mathbb{Z}^n , and always in the basis of fundamental weights.

Since the fundamental weights $\omega_1, \dots, \omega_n$ form a basis of $X(T)$, every weight $\omega \in X(T)$ can be written as $\omega = \sum_{i=1}^n m_i \omega_i$ for some $m_1, \dots, m_n \in \mathbb{Z}$. Computationally the weight ω is represented by the list of integers (m_1, \dots, m_n) . A weight $\omega = \sum_{i=1}^n m_i \omega_i$ is called *dominant* if $m_i \geq 0$ for all $i \in \{1, \dots, n\}$. There is a bijection between dominant weights of $X(T)$ and irreducible representations of G , which sends ω to the highest-weight representation $V(\omega)$. The identity element of $X(T)$, i.e., the zero element of \mathbb{Z}^n represented by the list $(0, \dots, 0)$, is a dominant weight, and it corresponds to the trivial representation of G .

A2. The special linear group. The special linear group $\mathrm{SL}_{n+1}(\mathbb{C})$ is the set of complex matrices with determinant 1; it is a simple Lie group of type A_n . The subset of diagonal matrices in $\mathrm{SL}_{n+1}(\mathbb{C})$ forms a subgroup which is a maximal torus of rank n . The corresponding Borel subgroup is the subset of upper-triangular matrices in $\mathrm{SL}_{n+1}(\mathbb{C})$.

The fundamental representations of $\mathrm{SL}_{n+1}(\mathbb{C})$ are (in order) the exterior powers $\mathbb{C}^{n+1}, \wedge^2 \mathbb{C}^{n+1}, \dots, \wedge^n \mathbb{C}^{n+1}$. Their weights are the fundamental weights $\omega_1, \dots, \omega_n$, and ω_i is represented by a list of zeroes with a 1 in the i -th entry.

The irreducible representations of $\mathrm{SL}_{n+1}(\mathbb{C})$ are given by the Schur modules $\mathbb{S}_\lambda \mathbb{C}^{n+1}$, where λ is a partition with at most n parts. For the construction of Schur modules the reader may consult [Fulton 1997, Chapter 8] or [Fulton and Harris 1991, Chapter 6]. The representation $\mathbb{S}_\lambda \mathbb{C}^{n+1}$ is the highest-weight representation $V(\omega)$ for the weight $\omega = (\lambda_1 - \lambda_2)\omega_1 + \dots + (\lambda_{n-1} - \lambda_n)\omega_{n-1} + \lambda_n\omega_n$.

To fix an example, consider the group $\mathrm{SL}_4(\mathbb{C})$ of type A_3 . The type of our group may be input by typing `DynkinType{"A",3}`. The representation \mathbb{C}^4 , also known as standard representation, has highest weight $\{1, 0, 0\}$, being the same as the Schur module $\mathbb{S}_1 \mathbb{C}^4$. The coordinate basis $\{e_1, e_2, e_3, e_4\}$ of \mathbb{C}^4 is a basis of weight vectors, and the weights of e_1, e_2, e_3, e_4 are $\{1, 0, 0\}$, $\{-1, 1, 0\}$, $\{0, -1, 1\}$ and $\{0, 0, -1\}$ respectively. If we tensor \mathbb{C}^4 with itself, we obtain a new representation of $\mathrm{SL}_4(\mathbb{C})$ with basis $e_i \otimes e_j$ for $i, j \in \{1, 2, 3, 4\}$. Moreover, the vector $e_i \otimes e_j$ is a weight vector with weight equal to the weight of e_i plus

the weight of e_j . The same principle of adding weights applies to symmetric and antisymmetric tensors, as well as to the tensor product of two different representations of the same group. For example, the weight of $e_1 \wedge e_2$ in $\wedge^2 \mathbb{C}^4$ is $\{0, 1, 0\}$. Indeed, $\wedge^2 \mathbb{C}^4$ is a highest-weight representation with highest weight $\{0, 1, 0\}$, thus corresponding to the Schur module $\mathbb{S}_{1,1} \mathbb{C}^4$, and $e_1 \wedge e_2$ is its highest-weight vector. Let us also mention the dual representation $(\mathbb{C}^4)^*$; the elements of the dual basis $e_1^*, e_2^*, e_3^*, e_4^*$ are weight vectors with weights $\{-1, 0, 0\}$, $\{1, -1, 0\}$, $\{0, 1, -1\}$ and $\{0, 0, 1\}$. Note that the weight of e_i^* is the additive inverse of the weight of e_i . Here the only highest weight is $\{0, 0, 1\}$, so $(\mathbb{C}^4)^*$ is isomorphic to $\wedge^3 \mathbb{C}^4$.

When dealing with a (quotient of a) product of groups, say $\mathrm{SL}_4(\mathbb{C}) \times \mathrm{SL}_6(\mathbb{C})$, we declare the type, listing in order the types of the simple components like this: `DynkinType{"A", 3}, {"A", 5}`. The representations of $\mathrm{SL}_4(\mathbb{C}) \times \mathrm{SL}_6(\mathbb{C})$ are obtained by tensoring a representation of $\mathrm{SL}_4(\mathbb{C})$ with one of $\mathrm{SL}_6(\mathbb{C})$. Their lists of weights are then concatenated to form a single list. For example, the representation $\mathbb{S}_{6,3,1} \mathbb{C}^4 \otimes \wedge^2 \mathbb{C}^6$ has highest weight $\{3, 2, 1, 0, 1, 0, 0, 0\}$, because the highest weights of $\mathbb{S}_{6,3,1} \mathbb{C}^4$ and $\wedge^2 \mathbb{C}^6$ are $\{3, 2, 1\}$ and $\{0, 1, 0, 0, 0\}$ respectively. Similarly, the irreducible representation with highest weight $\{0, 0, 0, 1, 1, 1, 1, 0\}$ is the tensor product of the irreducible representations \mathbb{C} of $\mathrm{SL}_4(\mathbb{C})$ and $\mathbb{S}_{4,3,2,1} \mathbb{C}^6$ of $\mathrm{SL}_6(\mathbb{C})$, which is simply isomorphic to $\mathbb{S}_{4,3,2,1} \mathbb{C}^6$.

ACKNOWLEDGEMENTS. The author wishes to thank Jianyun Shan for his help running the first example (mostly by hand!), Luke Oeding for submitting some interesting test examples and Adam Boocher for some helpful suggestions on the documentation. The author is indebted to the entire Macaulay2 community for its support; a special thanks goes to Mike Stillman and Dan Grayson, and to Baptiste Calmès and Viktor Petrov, whose Macaulay2 package *WeylGroups* facilitates many of the background computations carried out by our package. Finally, the author thanks the anonymous referees for providing many useful comments and helping to improve the quality of this article, as well as the source code and documentation of this package.

SUPPLEMENT. The online supplement contains version 0.6.5 of *HighestWeights*.

REFERENCES.

- [Eisenbud 1995] D. Eisenbud, *Commutative algebra*, Graduate Texts in Mathematics **150**, Springer, New York, 1995. MR 97a:13001 Zbl 0819.13001
- [Fulton 1997] W. Fulton, *Young tableaux*, London Mathematical Society Student Texts **35**, Cambridge University Press, 1997. MR 99f:05119 Zbl 0878.14034
- [Fulton and Harris 1991] W. Fulton and J. Harris, *Representation theory*, Graduate Texts in Mathematics **129**, Springer, New York, 1991. MR 93a:20069 Zbl 0744.22001

- [Galetto 2014] F. Galetto, “Free resolutions of orbit closures for the representations associated to gradings on Lie algebras of type E_6 , F_4 and G_2 ”, preprint, 2014. arXiv 1210.6410
- [Galetto 2015] F. Galetto, “Propagating weights of tori along free resolutions”, *J. Symb. Comput.* (2015). doi 10.1016/j.jsc.2015.05.004.
- [de Graaf 2000] W. A. de Graaf, *Lie algebras: theory and algorithms*, North-Holland Mathematical Library **56**, North-Holland, Amsterdam, 2000. MR 2001j:17011 Zbl 1122.17300
- [Grayson and Stillman] D. R. Grayson and M. E. Stillman, “Macaulay2: a software system for research in algebraic geometry”, available at <http://www.math.uiuc.edu/Macaulay2>.
- [Humphreys 1975] J. E. Humphreys, *Linear algebraic groups*, Graduate Texts in Mathematics **21**, Springer, New York-Heidelberg, 1975. MR 53 #633 Zbl 0325.20039
- [Humphreys 1978] J. E. Humphreys, *Introduction to Lie algebras and representation theory*, Graduate Texts in Mathematics **9**, Springer, New York-Berlin, 1978. Revised 2nd printing. MR 81b:17007 Zbl 0447.17001
- [Kreuzer and Robbiano 2005] M. Kreuzer and L. Robbiano, *Computational commutative algebra, II*, Springer, Berlin, 2005. MR 2006h:13036 Zbl 1090.13021
- [van Leeuwen et al. 1992] M. van Leeuwen, A. Cohen, and B. Lissier, “LiE, A Package for Lie Group Computations”, 1992, available at <http://www-math.univ-poitiers.fr/~maavl/LiE/>.
- [Moody and Patera 1982] R. V. Moody and J. Patera, “Fast recursion formula for weight multiplicities”, *Bull. Amer. Math. Soc. (N.S.)* **7**:1 (1982), 237–242. MR 84a:17005 Zbl 0494.17005
- [Sam 2009] S. V. Sam, “Computing inclusions of Schur modules”, *J. Softw. Algebra Geom.* **1** (2009), 5–10. MR 2878669 Zbl 1311.13039
- [Shafarevich 1994] I. R. Shafarevich, *Basic algebraic geometry, I: Varieties in projective space*, 2nd ed., Springer, Berlin, 1994. MR 95m:14001 Zbl 0797.14001
- [Tits 1967] J. Tits, *Tabellen zu den einfachen Lie Gruppen und ihren Darstellungen*, Springer, Berlin-New York, 1967. MR 36 #1575 Zbl 0166.29703

RECEIVED: 11 Jul 2014 REVISED: 23 Aug 2014 ACCEPTED: 5 Jun 2015

FEDERICO GALETTO:

galetto.federico@gmail.com

Department of Mathematics and Statistics, Queen’s University, 48 University Avenue,
Kingston K7L 3N6, Canada

Computing characteristic classes and the topological Euler characteristic of complex projective schemes

CHRISTINE JOST

ABSTRACT: The Macaulay2 package *CharacteristicClasses* provides commands for the computation of the topological Euler characteristic, the degrees of the Chern classes and the degrees of the Segre classes of a closed subscheme of complex projective space. The computations can be done both symbolically and numerically, the latter using an interface to Bertini. We provide some background for the implementation, show how to use the package with the help of examples and compare its performance to other implementations.

INTRODUCTION. The Macaulay2 [Grayson and Stillman] package *CharacteristicClasses* computes degrees of Chern and Segre classes of complex projective schemes. It also provides a command for computing the topological Euler characteristic. Recall that the Chern classes of a smooth scheme are defined to be the Chern classes of the tangent bundle. Moreover, the Segre classes of a scheme X embedded in \mathbb{P}^n are defined to be the Segre classes of the normal cone $C_X \mathbb{P}^n$ of the embedding. Note that the degree of a d -dimensional cycle class, i.e., the weighted sum of classes of d -dimensional subvarieties, is defined to be the weighted sum of the degrees of the subvarieties.

The computations done by the package *CharacteristicClasses* are based on a number of related algorithms which reduce the problem to the computation of the degrees of residuals. Briefly stated, residuals appear in the intersection of hypersurfaces containing a certain scheme. The intersection consists of this scheme and possibly other components, the residual. The algorithm computing Chern classes is described in [Di Rocco et al. 2011], the one computing Segre classes for possibly singular schemes in [Eklund et al. 2013], and the computation of the topological Euler characteristic is described in [Jost 2013]. The residuals can be computed either symbolically or numerically, which yields symbolic and numeric versions of

The work described in this article was performed while the author was at the University of Stockholm.
MSC2010: primary 14C17; secondary 14Q15, 65H10.

Keywords: Chern classes, Segre classes, topological Euler characteristic, computational algebraic geometry.

the algorithms. Both versions are implemented in the package, the numeric version via an interface to Bertini [Bates et al. 2013]. The algorithm is probabilistic, so with a small but nonzero probability the wrong result is obtained; see the next section or the documentation for more details.

There are other ways to compute the topological Euler characteristic and degrees of characteristic classes. The Macaulay2 command `euler` computes the topological Euler characteristic of smooth projective varieties by computing the Hodge numbers. Aluffi [2003] described algorithms for the computation of degrees of characteristic classes and the topological Euler characteristic. These algorithms are implemented in Aluffi's package *CSM*, which is not part of Macaulay2 but can be obtained at www.math.fsu.edu/~aluffi/CSM/CSM.html. The three aforementioned implementations complement each other, as is shown in [Eklund et al. 2013; Jost 2013] and shortly in the last section of this article. A new promising algorithm is described by Helmer [2015]. Helmer's algorithm is included in version 1.1 of *CharacteristicClasses*. This article describes version 1.0 of the package.

COMPUTING DEGREES OF CHERN AND SEGRE CLASSES. We describe how to make Chern and Segre classes of projective schemes computationally tractable by computing their degrees. The standard reference for the notions used in this section (Chern classes, Segre classes and Chow groups) is [Fulton 1984].

In this section, all schemes are defined over the field of complex numbers. Let X be a k -dimensional closed subscheme of \mathbb{P}^n , embedded by $i: X \hookrightarrow \mathbb{P}^n$, and denote its Chow group by $A_*(X) = \bigoplus_{d=0}^k A_d(X)$. Characteristic classes of X are elements of the graded group $A_*(X)$, whose generators are generally hard to compute. Hence algorithms for the computation of characteristic classes focus on computing a coarser invariant, the degrees of the classes. Let $\alpha = \sum_i \alpha_i [V_i]$ be a cycle class in $A_d(X)$, given as the weighted sum of classes of d -dimensional subvarieties V_i . Then the degree $\deg(\alpha)$ of α is defined to be the weighted sum $\sum_i \alpha_i \deg(V_i)$ of the degrees of the varieties V_i , seen as subvarieties of \mathbb{P}^n . One can also consider the pushforward of the cycle α to the Chow group of \mathbb{P}^n , which is known to be $A_*(\mathbb{P}^n) = \mathbb{Z}[H]/(H^{n+1})$, where H is the class of a general hyperplane. The two viewpoints are equivalent because $i_*(\alpha) = \deg(\alpha)H^{n-d}$.

The Chern classes $c_1(X), \dots, c_k(X)$ of a smooth scheme X are by definition the Chern classes $c_1(T_X), \dots, c_k(T_X)$ of the tangent bundle of X . The total Chern class is defined to be the sum $1 + c_1(X) + \dots + c_k(X)$. Furthermore, the Segre classes $s_1(X, \mathbb{P}^n), \dots, s_k(X, \mathbb{P}^n)$ of a possibly singular X are the Segre classes $s_1(C_X \mathbb{P}^n), \dots, s_k(C_X \mathbb{P}^n)$ of the normal cone $C_X \mathbb{P}^n$ of X in \mathbb{P}^n . If the embedding is regular, the normal cone is a vector bundle, called the normal bundle. The total Segre class is the sum $1 + s_1(X, \mathbb{P}^n) + \dots + s_k(X, \mathbb{P}^n)$.

CharacteristicClasses provides the commands `chernClass` and `segreClass`, which compute the pushforwards to the Chow group of \mathbb{P}^n of the total Chern and Segre classes of a given closed subscheme X of \mathbb{P}^n . This is equivalent to computing the degrees of the Chern and Segre classes. (To be precise, if the given scheme is singular, the command `chernClass` computes the degrees of the so-called Chern–Fulton class of the scheme, one of many generalizations of Chern classes to singular schemes. See [Fulton 1984, Example 4.2.6] or [Eklund et al. 2013, Remark 4.2] for more details on Chern–Fulton classes.) The closed subscheme X is given by the generators of a homogeneous ideal in a polynomial ring. The main idea of the algorithm is to relate the degrees of the Chern and Segre classes to the degrees of so-called residuals. One randomly chooses a number of hypersurfaces containing the scheme X , where the number of hypersurfaces is at least the codimension of X . According to a Bertini-type theorem, with probability 1 they intersect either in only X or in X and some components of expected codimension, the residual. Its degree can be computed either symbolically or numerically. Symbolically one computes the saturation using Gröbner basis techniques. The numeric computations can be done using software for the numeric solution of polynomial equation systems. *CharacteristicClasses* implements both the symbolic and numeric versions of the algorithms, using an interface to Bertini [Bates et al. 2013] for the latter. More details on the algorithms can be found in [Di Rocco et al. 2011] and [Eklund et al. 2013].

In practice, the probability that the intersection of hypersurfaces yields the correct residual is large but not 1. Hence the implementation may give the wrong result with a certain nonzero probability. An example of this phenomenon is given in the documentation of the package, accessed for instance by `viewHelp "probabilistic algorithm"`.

COMPUTING THE TOPOLOGICAL EULER CHARACTERISTIC. The topological Euler characteristic of a complex projective variety is the Euler characteristic of the underlying topological space with the usual Euclidean topology, i.e., the alternating sum of the Betti numbers. For the computations with *CharacteristicClasses*, we use that the topological Euler characteristic is equal to the degree of the top Chern–Schwartz–MacPherson class. More generally, the package can also compute the degrees of all the Chern–Schwartz–MacPherson classes. It uses the algorithm described in [Jost 2013], which reduces the computation of the degrees of Chern–Schwartz–MacPherson classes to the computation of Segre classes of singular subvarieties. For a more detailed introduction to Chern–Schwartz–MacPherson classes we refer to [Aluffi 2003] and [Jost 2013].

USING CHARACTERISTICCLASSES. We present four examples demonstrating how to use *CharacteristicClasses*. More detailed information on the commands

provided by this package can be found in its documentation, which is shown by for example using the command `viewHelp CharacteristicClasses`.

Example 1. We compute the degrees of the Chern classes of a toy example, the twisted cubic. We start by loading the package and defining the ideal of the twisted cubic, which is generated by the 2-by-2 minors of a 2-by-3 matrix. Then we use the command `chernClass` to obtain the pushforward of the total Chern class to the Chow ring of \mathbb{P}^n . The twisted cubic C_{tw} has dimension 1; hence it only has one Chern class $c_1(C_{\text{tw}})$. The total Chern class of the twisted cubic is $1 + c_1(C_{\text{tw}})$, and thus its pushforward to the Chow ring of the ambient space \mathbb{P}^3 is $\deg(C_{\text{tw}})H^2 + \deg(c_1(C_{\text{tw}}))H^3$, where H is the hyperplane class.

```
Macaulay2, version 1.6
with packages: ConwayPolynomials, Elimination, IntegralClosure,
               LLLBases, PrimaryDecomposition, ReesAlgebra, TangentCone

i1 : loadPackage "CharacteristicClasses";
--loading configuration for package "CharacteristicClasses" from
    file ../Macaulay2/init-CharacteristicClasses.m2

i2 : R = QQ[x,y,z,w];

i3 : twistedCubic = minors(2,matrix{{x,y,z},{y,z,w}})

o3 = ideal (- y^2 + x*z, - y*z + x*w, - z^2 + y*w)

o3 : Ideal of R

i4 : chernClass twistedCubic

o4 = 2H^3 + 3H^2

ZZ[H]
o4 : ----
      4
      H
```

So the above computations yield that $\deg(c_1(C_{\text{tw}})) = 2$ and confirms that the degree of the twisted cubic is $\deg(C_{\text{tw}}) = 3$. As the twisted cubic is smooth, the degree of the top Chern class equals its Euler characteristic $\chi(C_{\text{tw}})$, which is related to the genus $g(C_{\text{tw}})$ by $\chi = 2 - 2g$. As $\deg(c_1(C_{\text{tw}})) = \chi(C_{\text{tw}}) = 2$, this confirms that the genus is 0 and the twisted cubic is a rational curve.

All computations can also be done numerically using an interface to Bertini. Version 1.3 or higher of Bertini needs to be installed and the package must be configured correctly. For more information on the configuration, use `viewHelp "configuring Bertini"`. Then by using the value `Bertini` for the option `ResidualStrategy`, the computations are done numerically using Bertini, instead of via Gröbner basis computations.

```
i5 : chernClass(twistedCubic, ResidualStrategy=>Bertini)
```

$$o5 = 2H^3 + 3H^2$$

$$o5 : \frac{ZZ[H]}{H^4}$$

Example 2. We proceed with an example of the computation of Segre classes. The Whitney umbrella is a singular surface in \mathbb{P}^3 . Over the reals, it looks like an umbrella, and the singular locus like the handle of the umbrella. We compute the degree of the first Segre class $s_1(S, \mathbb{P}^n)$ of the singular locus S . Observe that in the following the singular locus is represented in Macaulay2 not as an ideal, but as a projective variety. All commands in this package work for both ideals and projective varieties.

```
i6 : whitney = ideal(x^2*w - y^2*z);
```

```
o6 : Ideal of R
```

```
i7 : handle = Proj singularLocus whitney;
```

```
i8 : segreClass handle
```

$$o8 = H^2$$

$$o8 : \frac{ZZ[H]}{H^4}$$

We see that the pushforward of the Segre class of the handle is $1 \cdot H^2 + 0 \cdot H^3$; hence $\deg(s_1(S, \mathbb{P}^n))$, the degree of its first Segre class, is 0. By [Aluffi 1994] the degree of the first Segre class is $\deg(s_1(S, \mathbb{P}^n)) = \nu - 2$, where ν is the number of pinch points of the surface. It follows that the Whitney umbrella has two pinch points. Only one of them is visible in the real patch $\{w \neq 0\}$ in which the Whitney umbrella is usually drawn; it is the tip of the umbrella.

Example 3. We use the Whitney umbrella from the above example to demonstrate the difference between different generalizations of Chern classes of smooth schemes to singular ones. *CharacteristicClasses* computes the degrees of two different generalizations, Chern–Schwartz–MacPherson classes and Chern–Fulton classes. The commands *CSMClass* and *chernClass* compute degrees of Chern–Schwartz–MacPherson and Chern–Fulton classes, respectively.

```
i9 : CSMClass whitney
```

$$o9 = 4H^3 + 4H^2 + 3H$$

$$o9 : \frac{ZZ[H]}{H^4}$$

```
i10 : chernClass whitney
```

$$o10 = 9H^3 + 3H^2 + 3H$$

$$o10 : \frac{ZZ[H]}{H^4}$$

For smooth schemes, the Chern–Schwartz–MacPherson classes and Chern–Fulton classes agree with the Chern classes. In the singular case, however, the two different generalizations disagree, as can be seen for instance from this example.

Example 4. We continue with an example from algebraic statistics, Example 2.2.2 of [Drton et al. 2009], which we will use to demonstrate computations of the topological Euler characteristic. The theory behind the computations is described in more detail in [Jost 2013]. The random censoring model with two events is a statistical model implicitly described by the ideal

$$(2p_0p_1p_2 + p_1^2p_2 + p_1p_2^2 - p_0^2p_{12} + p_1p_2p_{12})$$

in the polynomial ring $\mathbb{C}[p_0, p_1, p_2, p_{12}]$, where p_0, p_1, p_2 and p_{12} describe the probabilities of two events to occur before or after a third event. Given experimental data, one would like to compute the values for the probabilities p_0, p_1, p_2 and p_{12} which best describe the data, by maximizing the likelihood function. However, the function may have several stationary points, and methods such as the Newton method may only find a local maximum. Hence it makes sense to define the maximum likelihood degree as the number of critical points of the likelihood function, as was done in [Catanese et al. 2006]. By a theorem of Huh [2013], for a large class of examples the maximum likelihood degree equals the signed topological Euler characteristic of a certain open subvariety of the model. In this case, it is the topological Euler characteristic of the open subvariety

$$V(2p_0p_1p_2 + p_1^2p_2 + p_1p_2^2 - p_0^2p_{12} + p_1p_2p_{12}) \setminus V(p_0p_1p_2p_{12}(p_0 + p_1 + p_2 + p_{12})),$$

where $p_0p_1p_2p_{12} \neq 0$ means that no probability should be zero, and $p_0 + p_1 + p_2 + p_{12} \neq 0$ means that the probabilities should sum up to 1. We compute the

topological Euler characteristic using the command `eulerChar` together with the inclusion-exclusion principle.

```
i11 : S = QQ[p0,p1,p2,p12];
i12 : randomCensoring = ideal(2*p0*p1*p2 + p1^2*p2
                             + p1*p2^2 - p0^2*p12 + p1*p2*p12);
o12 : Ideal of S
i13 : boundary = ideal( p0*p1*p2*p12*(p0+p1+p2+p12) )
                  + randomCensoring;
o13 : Ideal of S
i14 : eulerChar randomCensoring
o14 = 5
i15 : eulerChar boundary
o15 = 2
```

It follows that the topological Euler characteristic of

$$V(2p_0p_1p_2 + p_1^2p_2 + p_1p_2^2 - p_0^2p_{12} + p_1p_2p_{12}) \setminus V(p_0p_1p_2p_{12}(p_0 + p_1 + p_2 + p_{12}))$$

is $5 - 2 = 3$. Hence the maximum likelihood degree of the random censoring model is 3, which confirms the result in [Drton et al. 2009].

PERFORMANCE. As stated in the introduction, the topological Euler characteristic can be computed by four different implementations, and degrees of Chern and Segre classes can be computed by three different implementations. The running times of these implementations, except Helmer’s 2014 implementation, are compared in more detail in [Eklund et al. 2013] and [Jost 2013]. Here, we briefly compare the running times of `CSMClass` in *CharacteristicClasses* to CSM in Aluffi’s implementation CSM (the December 2011 version), Helmer’s CSM in his prototype Macaulay2 implementation, and the routine `euler` from Macaulay2. Observe that the latter only works for nonsingular varieties and takes a projective variety as input. We use a 1.40 GHz two-core processor with 4 MB RAM. The results are summarized in Table 1.

All computations were done over the rational numbers. The ideal of the smooth surface is generated by the 2-by-2 minors of a 2-by-3 matrix of random linear forms.

As Table 1 shows, the Macaulay2 routine `euler`, the symbolic implementation by Aluffi and the one from this package complement each other. The performance of Helmer’s implementation is promising. The numeric implementation in this package is actually slower than the symbolic implementation for small examples like those shown in Table 1. However, for very large examples the symbolic

Input	CSMclass (symbolic)	CSMclass (numeric)	Aluffi's CSM	Helmer's CSM	euler
twisted cubic	< 1	47	2	< 1	< 1
smooth surface in \mathbb{P}^4 defined by minors	34	2131	14777	3	88
Segre embedding of $\mathbb{P}^1 \times \mathbb{P}^2$ in \mathbb{P}^5	10	285	1	1	< 1

Table 1. Comparison of run times (all times given in seconds).

methods may not terminate due to insufficient memory; the numeric implementation does not have this problem.

ACKNOWLEDGEMENTS. Many thanks to the organizers and participants of the 2011 Macaulay2 workshop at the IMA in Minneapolis, where parts of the package were written. Also many thanks to my adviser Sandra Di Rocco for help with the article.

Jon Hauenstein adapted the output of the regenerative cascade in Bertini to the interface used in *CharacteristicClasses*. The example of the computation of the number of pinch points of the Whitney umbrella is due to David Eklund.

Finally I would like to thank the referees for careful reading and much helpful advice.

REFERENCES.

- [Aluffi 1994] P. Aluffi, “MacPherson’s and Fulton’s Chern classes of hypersurfaces”, *Internat. Math. Res. Notices* 11 (1994), 455–465. MR 96d:14004 Zbl 0839.14035
- [Aluffi 2003] P. Aluffi, “Computing characteristic classes of projective schemes”, *J. Symbolic Comput.* **35**:1 (2003), 3–19. MR 2004b:14007 Zbl 1074.14502
- [Bates et al. 2013] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler, “Bertini: Software for Numerical Algebraic Geometry”, 2013, Available at <http://www.nd.edu/~sommese/bertini>.
- [Catanese et al. 2006] F. Catanese, S. Hoşten, A. Khetan, and B. Sturmfels, “The maximum likelihood degree”, *Amer. J. Math.* **128**:3 (2006), 671–697. MR 2007m:13036 Zbl 1123.13019
- [Di Rocco et al. 2011] S. Di Rocco, D. Eklund, C. Peterson, and A. J. Sommese, “Chern numbers of smooth varieties via homotopy continuation and intersection theory”, *J. Symbolic Comput.* **46**:1 (2011), 23–33. MR 2012h:14010 Zbl 1200.14014
- [Drton et al. 2009] M. Drton, B. Sturmfels, and S. Sullivant, *Lectures on algebraic statistics*, Oberwolfach Seminars **39**, Birkhäuser, Basel, 2009. MR 2012d:62004 Zbl 1166.13001
- [Eklund et al. 2013] D. Eklund, C. Jost, and C. Peterson, “A method to compute Segre classes of subschemes of projective space”, *J. Algebra Appl.* **12**:2 (2013), 1250142, 15. MR 3005594 Zbl 1274.13044
- [Fulton 1984] W. Fulton, *Intersection theory*, Ergebnisse der Mathematik und ihrer Grenzgebiete (3) **2**, Springer, Berlin, 1984. MR 85k:14004 Zbl 0541.14005

- [Grayson and Stillman] D. R. Grayson and M. E. Stillman, “Macaulay2: a software system for research in algebraic geometry”, Available at <http://www.math.uiuc.edu/Macaulay2>.
- [Helmer 2015] M. Helmer, “Algorithms to compute the topological Euler characteristic, Chern–Schwartz–MacPherson class and Segre class of projective varieties”, *J. Symbolic Comput.* (online publication March 2015).
- [Huh 2013] J. Huh, “The maximum likelihood degree of a very affine variety”, *Compos. Math.* **149**:8 (2013), 1245–1266. MR 3103064 Zbl 1282.14007
- [Jost 2013] C. Jost, “An algorithm for computing the topological Euler characteristic of complex projective varieties”, preprint, 2013. arXiv 1301.4128

RECEIVED: 11 Jul 2014

ACCEPTED: 5 Jun 2015

CHRISTINE JOST:

christine.jost@ericsson.com

Ericsson Research, Färögatan 6, SE-164 80 Stockholm, Sweden

<i>Software for multiplier ideals</i>	1
Zach Teitler	
<i>Partially ordered sets in Macaulay2</i>	9
David Cook II, Sonja Mapes and Gwyneth Whieldon	
<i>Free resolutions and modules with a semisimple Lie group action</i>	17
Federico Galetto	
<i>Computing characteristic classes and the topological Euler characteristic of complex projective schemes</i>	31
Christine Jost	

