

```

gap> g:= SymmetricGroup( 4 )
Sym( [ 1 .. 4 ] )
gap> tbl:= CharacterTable( g ); HasIrr( tbl );
15 : betti(t,Weights=>{1,0})
false
0 1 2 3 4 gap> tblmod2:= CharacterTable( tbl, 2 );
o5 = total: 1 4 13 14 4 BrauerTable( Sym( [ 1 .. 4 ] ), 2 )
0: 1 . . . .
1: . 2 2 4 2 gap> tblmod2 = CharacterTable( tbl, 2 );
2: . 2 5 6 . true
3: . . 4 . 2
4: . . . 4 . gap> tblmod2 = BrauerTable( tbl, 2 );
5: . . 2 . . true
o5 : BrauerTable( Sym( [ 1 .. 4 ] ), 2 )
16 : betti(t,Weights=>{0,1})
0 1 2 3 4 gap> libtbl:= CharacterTable( "M" );
o6 = total: 1 4 13 14 4 CharacterTable( "M" )
0: 1 . . . . gap> CharacterTableRegular( libtbl, 2 );
1: . 2 . . 2 BrauerTable( "M" )
2: . 2 . . 2 gap> BrauerTable( libtbl, 2 );
3: . . 4 . 2 fail
4: . . . 4 . ring r1 = 32003,(x,y,z),ds;
5: . . 2 . . gap> CharacterTable( "Symmetric", 4 );
o6 : BettiTally CharacterTable( "Sym(4)" )
int a,b,c,t=11,5,3,0;
17 : t1 = betti(t,Weights=>{1,1}) 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;
gap> ComputedBrauerTables( tbl );
[ , BrauerTable( Sym( [ 1 .. 4 ] ), 2 ) ]
option(noprot);
o7 = total: 1 4 13 14 4 timer=1;
0: 1 . . . . ring r2 = 32003,(x,y,z),dp;
1: . . . . poly f=imap(r1,f);
2: . . . . ideal j=jacob(f);
3: . 2 . . vdim(std(j));
4: . . . . ==> 536
5: . 2 . . vdim(std(j+f));
6: . . 1 . . ==> 195
7: . . 8 6 . timer=0; // reset timer
8: . . 4 8 4
o7 : BettiTally
18 : 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, 4}, 11) => 6
(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

```

Journal of Software for Algebra and Geometry

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

CHRISTINE JOST

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 (- y2 + x*z, - y*z + x*w, - z2 + y*w)
```

```
o3 : Ideal of R
```

```
i4 : chernClass twistedCubic
```

```
o4 = 2H3 + 3H2
```

```
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)
      3      2
o5 = 2H  + 3H
      ZZ[H]
o5 : -----
      4
      H
```

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
      2
o8 = H
      ZZ[H]
o8 : -----
      4
      H
```

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](http://www.math.uiuc.edu/Macaulay2)”, 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> Zach Teitler	1
<i>Partially ordered sets in Macaulay2</i> David Cook II, Sonja Mapes and Gwyneth Whieldon	9
<i>Free resolutions and modules with a semisimple Lie group action</i> Federico Galetto	17
<i>Computing characteristic classes and the topological Euler characteristic of complex projective schemes</i> Christine Jost	31