

```

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
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
gap> tblmod2 = BrauerTable( tbl, 2 );
true
o5 : BettiTable
16 : betti(t,Weights=>{0,1})
0 1 2 3 4
o6 = total: 1 4 13 14 4
0: 1 . . . .
1: . 2 . . 2
2: . 2 . . 2
3: . 4 . . 2
4: . . . 4 .
5: . . 2 . .
gap> libtbl:= CharacterTable( "M" );
CharacterTable( "M" )
gap> CharacterTableRegular( libtbl, 2 );
BrauerTable( "M", 2 )
gap> BrauerTable( libtbl, 2 );
fail
gap> CharacterTable( "Symmetric", 4 );
int a,b,c,t=11,5,3,0;
CharacterTable( "Sym(4)" )
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);
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
o6 : BettiTally
17 : t1 = betti(t,Weights=>{1,1})
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
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, 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

```

Journal of Software for Algebra and Geometry

A Macaulay2 package for computations with rational maps
 GIOVANNI STAGLIANÒ

A Macaulay2 package for computations with rational maps

GIOVANNI STAGLIANÒ

ABSTRACT: The Macaulay2 package `Cremona.m2` performs some computations on rational and birational maps between irreducible projective varieties. For instance, it provides methods to compute degrees and projective degrees of rational maps without any theoretical limitation, from which is derived a general method to compute the push-forward to projective space of Segre classes. Moreover, the computations can be done both deterministically and probabilistically. We give here a brief description of the methods and algorithms implemented.

INTRODUCTION. In this note we describe the computational package `Cremona.m2`, included with [Macaulay2] since version 1.9. A first rudimentary version of this package has been already used in an essential way in [Staglianò 2016] (it was originally named `bir.m2`), and recent applications can be found in [Staglianò 2018; Russo and Staglianò 2017]. Here we describe version 4.2.2 of the package.

`Cremona.m2` performs computations on rational and birational maps between absolutely irreducible projective varieties over a field \mathbb{K} . Among other things, it provides general methods to compute projective degrees of rational maps, from which, as is well known (see Proposition 1.2), one can interpret them as methods to compute the push-forward to projective space of Segre classes. The algorithms are naively derived from the mathematical definitions, with the advantages of being obvious, quite general and easily implemented. Moreover, all the methods (where this may make sense) are available both in a probabilistic version and in a deterministic version, and one can switch from one to the other with a boolean option named `MathMode`.

In Section 1, we will describe the main methods provided by the package and the algorithms implemented. Most of these have already been described in [Staglianò 2016, Section 2], but here we will consider a more general setting. For instance, Algorithm 1.3 for computing homogeneous components of kernels of homogeneous ring maps was presented in [Staglianò 2016, Algorithm 2.5] requiring that

MSC2010: 14E05, 14Q15.

Keywords: rational map, birational map, projective degrees, Segre class.

Cremona.m2 version 4.2.2

the map was between polynomial rings. In [Section 2](#), we will show how these methods work in some particular examples, concluding with an experimental comparison of the running times of one of these methods with the corresponding ones proposed in [[Helmer 2016](#)] and [[Harris 2017](#)] (see also [[Jost 2015](#)]). For further technical details we refer to the documentation of the package, which can be shown using the command `viewHelp Cremona`.

We mention that the package `RationalMaps.m2`, by K. Schwede, D. Smolkin, S. H. Hassanzadeh, and C. J. Bott, is another package included with Macaulay2 for working with rational maps. It mainly focuses on providing a general method for inverting birational maps, which in some cases turns out to be competitive with the corresponding method of `Cremona.m2`.

1. DESCRIPTION OF THE MAIN METHODS. Throughout, we shall use the following notation. Let \mathbb{K} denote a field; in practice, it can be for instance \mathbb{Q} , a finite field, or a fraction field of a polynomial ring over these. Let $\phi : X \dashrightarrow Y$ be a rational map from a subvariety $X = V(I) \subseteq \mathbb{P}^n = \text{Proj}(\mathbb{K}[x_0, \dots, x_n])$ to a subvariety $Y = V(J) \subseteq \mathbb{P}^m = \text{Proj}(\mathbb{K}[y_0, \dots, y_m])$, which can be represented, although not uniquely, by a homogeneous ring map

$$\varphi : \mathbb{K}[y_0, \dots, y_m]/J \rightarrow \mathbb{K}[x_0, \dots, x_n]/I$$

of quotients of polynomial rings by homogeneous ideals. Sometimes we will denote by $F_0, \dots, F_m \in \mathbb{K}[x_0, \dots, x_n]$ homogeneous forms of the same degree such that $\overline{F}_i := F_i + I = \varphi(y_i)$, for $i = 0, \dots, m$. The common degree of these elements will be denoted by δ .

From algebraic geometry to computational algebra. For each homogeneous ideal $\mathfrak{a} \subseteq \mathbb{K}[x_0, \dots, x_n]/I$ (resp. $\mathfrak{b} \subseteq \mathbb{K}[y_0, \dots, y_m]/J$), we have a closed subscheme $V(\mathfrak{a}) \subseteq X$ (resp. $V(\mathfrak{b}) \subseteq Y$), and the following basic formulae hold:¹

$$\overline{\phi(V(\mathfrak{a}))} = V(\varphi^{-1}(\mathfrak{a})) \quad \text{and} \quad \overline{\phi^{-1}(V(\mathfrak{b}))} = V((\varphi(\mathfrak{b})) : (\varphi(y_0), \dots, \varphi(y_m))^\infty). \quad (1-1)$$

In particular, the (closure of the) image of ϕ is defined by the kernel of φ . Several issues concerning rational maps lead naturally to an examination of the left-hand sides of (1-1), and the right-hand sides of (1-1) can be determined using Gröbner basis techniques, whenever \mathfrak{a} and \mathfrak{b} are explicitly given. Furthermore, Macaulay2 provides useful commands such as `preimage`, `kernel` and `saturate`, so that the required user skill level is quite low. The aim of the package `Cremona.m2` is to provide further tools.

¹By abuse of notation, we consider ϕ as a morphism defined on the open set $X \setminus V(\overline{F}_0, \dots, \overline{F}_m)$.

Computing projective degrees. The projective degrees are the most basic invariants of a rational map. Many others can be derived from them, such as, for instance, the dimension and the degree of the base locus. For more details on the subject, see [Harris 1992, Example 19.4, p. 240].

Definition 1.1 (projective degrees, [Harris 1992]). (1) The projective degrees $d_0(\phi), d_1(\phi), \dots, d_{\dim X}(\phi)$ of the map ϕ are defined as the components of the multidegree of the closure of the graph $\Gamma_\phi \subset \mathbb{P}^n \times \mathbb{P}^m$.

(2) Equivalently, the i -th projective degree $d_i(\phi)$ can be defined in terms of dimension and degree of the closure of $\phi^{-1}(L)$, where L is a general $(m - \dim X + i)$ -dimensional linear subspace of \mathbb{P}^m ; more precisely, $d_i(\phi) = \deg \phi^{-1}(L)$ if $\dim \phi^{-1}(L) = i$, and $d_i(\phi) = 0$ otherwise.

In common computer algebra systems such as Macaulay2, it is easy to translate Definition 1.1 into code. We now describe in more detail how this can be done. All of this is implemented in the method `projectiveDegrees`; see Example 2.2 for an example using it.

Deterministic approach. Taking into account Definition 1.1(1), a bihomogeneous ideal for Γ_ϕ in $\mathbb{K}[x_0, \dots, x_n, y_0, \dots, y_m]$ can be, for instance, obtained as

$$(I + (\{y_i F_j - y_j F_i, 0 \leq i, j \leq m\})) : (F_0, \dots, F_m)^\infty. \quad (1-2)$$

Therefore its multidegree can be computed in Macaulay2 with `multidegree`, which implements an algorithm according to [Miller and Sturmfels 2005, p. 165].

Probabilistic approach. (See also [Staglianò 2016, Remark 2.4].) Taking into account Definition 1.1(2), if L is defined by an ideal I_L , the second formula of (1-1) tells us that $\phi^{-1}(L)$ is defined by the saturation of the ideal $(\varphi(I_L))$ by $(\overline{F}_0, \dots, \overline{F}_m)$ in the ring $\mathbb{K}[x_0, \dots, x_n]/I$. So replacing the word *general* with *random* in the definition, we get a probabilistic algorithm that computes all the projective degrees. Moreover, we can considerably speed up this algorithm by taking into account two remarks: firstly, the saturation

$$\varphi(I_L) : (\overline{F}_0, \dots, \overline{F}_m)^\infty$$

is the same as

$$\varphi(I_L) : (\lambda_0 \overline{F}_0 + \dots + \lambda_m \overline{F}_m)^\infty,$$

where $\lambda_0, \dots, \lambda_m \in \mathbb{K}$ are general scalars; secondly, the i -th projective degree of ϕ coincides with the $(i-1)$ -th projective degree of the restriction of ϕ to a general hyperplane section of X .

An alternative deterministic approach. Replacing the word *general* with *symbolic* in Definition 1.1(2) gives us a deterministic algorithm for computing projective degrees. For instance, in the case in which $\phi : \mathbb{P}^n \dashrightarrow \mathbb{P}^n$ is a dominant rational

map, extending \mathbb{K} to the fractional field of a polynomial ring $\mathbb{K}[a_0, \dots, a_n]$, we have that $d_0(\phi)$ is the degree of the fiber of ϕ at the *symbolic point* $[a_0, \dots, a_n]$.

Some applications using projective degrees.

The degree of a rational map. The degree of the map $\phi : X \dashrightarrow Y$ is the number of isolated points in the inverse image of a general point of $\overline{\phi(X)}$ over the algebraic closure of \mathbb{K} . This is the same as the ratio of $d_0(\phi)$ and $\deg \overline{\phi(X)}$, and thus it can be explicitly computed. Let us note, however, that in several cases we do not need to compute the kernel of ϕ . For instance, if X is a projective space, we are able to pick an abundance of rational points of $\overline{\phi(X)}$ and then we apply the second formula of (1-1). Another special case is when $d_0(\phi)$ is a prime number: here we have only to establish if the image of ϕ is a linear subspace (e.g., applying [Algorithm 1.3](#) with $d = 1$). The method provided by `Cremona.m2` for this computation is named `degreeOfRationalMap`.²

Methods related to this are `isBirational` and `isDominant`, with obvious meaning. The latter does not compute the kernel of ϕ , but it uses an algorithm that looks for $d_r(\phi)$, where $r = \dim X - \dim Y$. More precisely, the algorithm is based on the following fact: let $Z \subset Y$ be a random 0-dimensional linear section of Y ; if $\dim \overline{\phi^{-1}(Z)} = \dim X - \dim Y \geq 0$, then ϕ is certainly dominant, otherwise it is probably not dominant (see [[Mumford 1988](#), Chapter I, § 8] or [[Hartshorne 1977](#), Chapter II, Exercise 3.22]). When this last case occurs, it is generally easy to find a nonzero element in the kernel of ϕ , and so this method turns out to be very effective even in its deterministic version (see [Example 2.1](#)).

The Segre class. It is well known that one can deduce an algorithm computing the push-forward to projective space of Segre classes from an algorithm computing projective degrees of rational maps between projective varieties and vice versa. Indeed, with our notation, we have the following:

Proposition 1.2 ([[Fulton 1984](#), Proposition 4.4]; see also [[Dolgachev 2011](#), Section 2.3; [Aluffi 2003](#), Section 3]). *Let $\mathfrak{B} \subset X$ be the subscheme defined by $\overline{F_0}, \dots, \overline{F_m}$ and let $\nu : X \hookrightarrow \mathbb{P}^n$ be the inclusion. If H denotes the hyperplane class of \mathbb{P}^n and $r = \dim X$, then the push-forward $\nu_*(s(\mathfrak{B}, X))$ of the Segre class of \mathfrak{B} in X is*

$$\nu_*(s(\mathfrak{B}, X)) = \sum_{k=0}^{\dim \mathfrak{B}} ((-1)^{r-k-1} \sum_{i=0}^{r-k} (-1)^i \binom{r-k}{i} \delta^{r-k-i} d_{r-i}(\phi)) H^{n-k}. \quad (1-3)$$

The general method `SegreClass`, provided by `Cremona.m2` for computing the push-forward to projective spaces of Segre classes, does basically nothing more

²Notice that, in general, if the result of the probabilistic algorithm for `degreeOfRationalMap` is wrong, it can be either too small or too large. However, as a consequence of [[Hartshorne 1977](#), Chapter III, Exercise 10.9], it should always provide a lower bound when the map is dominant between smooth varieties.

than apply (1-3); see [Example 2.3](#) for an example using this method. Furthermore, applying one of the main results in [\[Aluffi 2003\]](#), a method is derived to compute the push-forward to projective space of the Chern–Schwartz–MacPherson class $\text{CSM}(W)$ of the support of a projective scheme W ; recall that the component of dimension 0 of $\text{CSM}(W)$ is the topological Euler characteristic of the support of W .

Computing homogeneous components of kernels. To compute, using Macaulay2, the homogeneous component of degree d of the kernel of phi ($= \varphi$), one can perform the command `ideal image basis(d, kernel phi)`. This is equivalent to the command `kernel(phi, d)` provided by `Cremona.m2`, but the latter uses the following obvious algorithm.

Algorithm 1.3. **Input:** the ring map φ and an integer d .

Output: homogeneous component of degree d of the kernel of φ .

- Find vector space bases G_0, \dots, G_r of $(\mathbb{K}[y_0, \dots, y_m]/J)_d$ and H_0, \dots, H_s of $I_{d\delta}$, where subscripts stand for homogeneous components.
- Take generic linear combinations $\mathbf{G} = \sum_{i=0}^r a_i G_i$ and $\mathbf{H} = \sum_{j=0}^s b_j H_j$, and find a basis of solutions for the homogeneous linear system obtained by requiring that the polynomial

$$\mathbf{G}(F_0, \dots, F_m) - \mathbf{H} \in \mathbb{K}[a_0, \dots, a_r, b_0, \dots, b_s][x_0, \dots, x_n]$$

vanishes identically.

- For each vector $(\hat{a}_0, \dots, \hat{a}_r, \hat{b}_0, \dots, \hat{b}_s) \in \mathbb{K}^{r+s+2}$ obtained in the previous step, replace in \mathbf{G} the coefficients a_0, \dots, a_r with $\hat{a}_0, \dots, \hat{a}_r$; return all these elements.

For small values of d , applying [Algorithm 1.3](#) may turn out to be much faster than computing a list of generators of the kernel of the map; see for instance [Example 2.1](#) below.

Inverting birational maps. General algorithms for inverting birational maps are known. One of them is implemented in the package [Parametrization.m2](#) by J. Boehm, and the method `inverseMap` of `Cremona.m2` uses the same one for the general case as well. However, when the source X of the rational map ϕ is a projective space and a further technical condition is satisfied, then it uses the following powerful algorithm.

Algorithm 1.4 ([\[Russo and Simis 2001\]](#); see also [\[Simis 2004\]](#)). **Input:** the ring map φ (assuming that ϕ is birational and further conditions are satisfied).

Output: a ring map representing the inverse map of ϕ .

- Find generators $\{(L_{0,j}, \dots, L_{m,j})\}_{j=1,\dots,q}$ for the module of linear syzygies of (F_0, \dots, F_m) .
- Compute the Jacobian matrix Θ of the bihomogeneous forms

$$\left\{ \sum_{i=0}^m y_i L_{i,j} \right\}_{j=1,\dots,q}$$

with respect to the variables x_0, \dots, x_n and consider the map of graded free modules $(\Theta \bmod J) : (k[y_0, \dots, y_m]/J)^{n+1} \rightarrow (k[y_0, \dots, y_m]/J)^q$.

- Return the map defined by a generator $G = (G_0, \dots, G_n)$ for the kernel of $(\Theta \bmod J)$.

Remark 1.5. One of the main features of the package `RationalMaps.m2`, by Schwede, Smolkin, Hassanzadeh, and Bott, is a method for inverting birational maps, which, in the case when [Algorithm 1.4](#) does not apply, appears to be quite competitive with the method `inverseMap` of `Cremona.m2`.

Heuristic approach. The method `approximateInverseMap` provides a heuristic approach to compute the inverse of a birational map modulo a change of coordinate. The idea of the algorithm is to try to construct the base locus of the inverse by looking for the images of general linear sections. Consider, for simplicity, the case in which $\phi : \mathbb{P}^n \dashrightarrow \mathbb{P}^{n'}$ is a Cremona transformation. Then, by taking the images of $n+1$ general hyperplanes in \mathbb{P}^n , we form a linear system of hypersurfaces in $\mathbb{P}^{n'}$ of degree $d_1(\phi)$ which defines a rational map $\psi : \mathbb{P}^{n'} \dashrightarrow \mathbb{P}^n$ such that $\psi \circ \phi$ is a (linear) isomorphism; i.e., we find an *approximation* of ϕ^{-1} . Next, we can fix the *error of the approximation* by observing that we have $\phi^{-1} = (\psi \circ \phi)^{-1} \circ \psi$. It is surprising that this method turns to be effective in examples where other deterministic algorithms seem to run endlessly; see for instance [Example 2.1](#) below.

2. EXAMPLES. In this section, we show how the methods described in [Section 1](#) can be applied in some particular examples. We note that the package `Cremona.m2` provides the data type `RationalMap`, but here we will use the more familiar type `RingMap`. For brevity, we will omit irrelevant output lines. We start with an example reviewing the construction given in [\[Staglianò 2016\]](#) of a quadro-quadric Cremona transformation of \mathbb{P}^{20} .

Example 2.1. The code below constructs a ring map `psi` representing a rational map $\psi : \mathbb{P}^{16} \dashrightarrow \mathbb{P}^{20}$. Precisely, the algorithm for constructing ψ is as follows: take $E \subset \mathbb{P}^7$ to be a 3-dimensional edge variety of degree 7, namely, the residual intersection of $\mathbb{P}^1 \times \mathbb{P}^3 \subset \mathbb{P}^7$ with a general quadric in \mathbb{P}^7 containing one of the \mathbb{P}^3 's of the rulings of $\mathbb{P}^1 \times \mathbb{P}^3 \subset \mathbb{P}^7$; next, see $E \subset \mathbb{P}^7$ embedded in a hyperplane of \mathbb{P}^8 and take the birational map $\phi : \mathbb{P}^8 \dashrightarrow \mathbb{P}^{16}$ defined by the quadrics of \mathbb{P}^8 containing E ; take $\psi : \mathbb{P}^{16} \dashrightarrow \mathbb{P}^{20}$ to be the map defined by the quadrics of \mathbb{P}^{16}

containing the image of ϕ . For the first part of this construction, we use the package `Cremona.m2` only to shorten the code.

```
Macaulay2, version 1.11
with packages: ConwayPolynomials, Elimination, IntegralClosure, InverseSystems,
              LLLBases, PrimaryDecomposition, ReesAlgebra, TangentCone
i1 : loadPackage "Cremona";
i2 : K = ZZ/70001;
i3 : P8 = K[t_0..t_8]; E = saturate(minors(2, genericMatrix(P8, 4, 2)) + sum(
      (ideal(t_0..t_7)*ideal(t_0..t_3))_* , u->random(K)*u), ideal(t_0..t_3)) + t_8;
i5 : psi = toMap kernel(toMap(E, 2), 2);
```

Up to this point, the computation was standard. But now we want to determine the homogeneous ideal of $Z := \overline{\psi(\mathbb{P}^{16})} \subset \mathbb{P}^{20}$, which turns out to be generated by quadrics. Computing this using `kernel psi` seems an impossible task, but it is elementary using `kernel(psi, 2)`. So we can consider ψ as a dominant rational map $\psi : \mathbb{P}^{16} \dashrightarrow Z \subset \mathbb{P}^{20}$.

```
i6 : time Z = kernel(psi, 2);
      -- used 2.84998 seconds
i7 : psi = toMap(psi, Dominant=>Z);
```

The map ψ turns out to be not only dominant but birational.

```
i8 : time degreeOfRationalMap psi
      -- used 2.11216 seconds
o8 = 1
```

We now want to compute the inverse of ψ . This is a case where `inverseMap` can apply [Algorithm 1.4](#), but the running time is several hours. We can perform this computation in seconds by using `approximateInverseMap`.

```
i9 : time psi' = approximateInverseMap(psi, CodimBsInv=>10, MathMode=>true);
      -- used 15.9724 seconds
```

A Cremona transformation ω of \mathbb{P}^{20} is then obtained combining ψ^{-1} and Z as follows.

```
i10 : omega = toMap(lift(matrix psi', ring Z)|gens Z);
```

Even checking just the dominance of ω , by computing `kernel omega`, seems an impossible task, but it can be done quickly with `isDominant`.

```
i11 : time isDominant(omega, MathMode=>true)
      -- used 0.100369 seconds
o11 = true
```

We now check that our map is birational and find its inverse using [Algorithm 1.4](#).

```
i12 : time isBirational omega
      -- used 0.0366468 seconds
o12 = true
i13 : time inverseMap omega;
      -- used 0.0717518 seconds
```

Example 2.2. Here, we use the probabilistic versions of some methods. Take M to be a generic 3×5 matrix of linear forms on \mathbb{P}^6 , and, let $\phi : \mathbb{P}^6 \dashrightarrow \mathbb{G}(2, 4) \subset \mathbb{P}^9$

be the rational map defined by the 3×3 minors of M (its base locus is a smooth threefold scroll over a plane).

```
i14 : P6 = K[x_0..x_6]; M = matrix pack(5,for i from 1 to 15 list random(1,P6));
i16 : phi = toMap(minors(3,M),Dominant=>2);
```

We check that the map is birational and compute its inverse.

```
i17 : time isBirational phi
      -- used 0.217607 seconds
o17 = true
i18 : time psi = inverseMap phi;
      -- used 1.39511 seconds
```

Now we compute the multidegrees of ϕ and ϕ^{-1} .

```
i19 : time (projectiveDegrees phi, projectiveDegrees psi)
      -- used 1.37582 seconds
o19 = ({1, 3, 9, 17, 21, 15, 5}, {5, 15, 21, 17, 9, 3, 1})
```

We also compute the push-forward to \mathbb{P}^6 (resp. \mathbb{P}^9) of the Segre class of the base locus of ϕ (resp. ϕ^{-1}) in \mathbb{P}^6 (resp. in $\mathbb{G}(2, 4)$). As usual, H denotes the hyperplane class.

```
i20 : time (SegreClass phi, SegreClass psi)
      -- used 1.43359 seconds
o20 = (- 680H6 + 228H5 - 60H4 + 10H3 , 728H9 - 588H8 + 276H7 - 98H6 + 24H5)
```

Example 2.3. In this example, we use the deterministic version of the method `SegreClass`. We take $Y \subset \mathbb{P}^{11}$ to be the dual quartic hypersurface of

$$\mathbb{P}^1 \times Q^4 \subset \mathbb{P}^{11*},$$

where $Q^4 \subset \mathbb{P}^5$ is a smooth quadric hypersurface, and take $X \subset Y$ to be the singular locus of Y . We then compute the push-forward to the Chow ring of \mathbb{P}^{11} of the Segre class both of X in Y and of X in \mathbb{P}^{11} working over the Galois field $\text{GF}(331^2)$.

```
i21 : P11 = GF(331^2)[x_0..x_11];
i22 : Y = ideal sum(first entries gens minors(2,genericMatrix(P11,6,2)),t->t^2);
i23 : X = sub(ideal jacobian Y,P11/Y);
i24 : time SegreClass(X, MathMode=>true) -- push-forward of s(X,Y)
      -- used 0.789986 seconds
o24 = 507384H11 - 137052H10 + 35532H9 - 9018H8 + 2340H7 - 658H6 + 204H5 - 64H4 + 16H3
i25 : time SegreClass(lift(X,P11), MathMode=>true) -- push-forward of s(X,P^11)
      -- used 0.846234 seconds
o25 = 313568H11 - 101712H10 + 30636H9 - 8866H8 + 2532H7 - 720H6 + 198H5 - 48H4 + 8H3
```

Example 2.4. Here we experimentally measure the probability of obtaining an incorrect answer using the probabilistic version of the method `projectiveDegrees` with a simple example of a birational map $\phi : \mathbb{G}(1, 3) \dashrightarrow \mathbb{P}^4$ defined over \mathbb{K} . We define a procedure which computes this probability as a function of the field \mathbb{K} .

field	\mathbb{Q}	$\mathbb{Z}/70001$	$\text{GF}(3^8)$	$\mathbb{Z}/101$	$\mathbb{Z}/31$
probability	0.0	0.0	0.002	0.074	0.253

Table 1. Incorrect outputs of a probabilistic method.

In [Table 1](#), we report the results obtained by running the procedure with various fields.

```
i26 : p = (K) -> (
  x := local x; R := K[x_0..x_4];
  phi := inverseMap toMap(minors(2,matrix{{x_0..x_3},{x_1..x_4}}),Dominant=>2);
  m := projectiveDegrees(phi,MathMode=>true);
  0.1 * # select(1000,i -> projectiveDegrees phi != m);
```

Example 2.5. Lastly, we deal with an experimental comparison of the method `SegreClass` of `Cremona.m2` and the corresponding ones of other Macaulay2 packages. Precisely, we want to compare the method `SegreClass` against the corresponding methods of the packages `CharacteristicClasses.m2` version 2.0, by M. Helmer and C. Jost (see [\[Helmer 2016; Jost 2015\]](#)), which provides a probabilistic method; and `FMPIntersectionTheory.m2` version 0.1, by C. Harris [\[2017\]](#), which provides a deterministic method. Since the former puts restrictions on the ambient variety, we will only consider examples where the ambient is a projective space. We are unable to determine precisely which is the fastest among all the methods and which, in the probabilistic case, has highest probability of giving the correct answer. We just summarize in [Table 2](#) the running times for some special examples. Below is the code from which we obtained the first row of the table.

```
i27 : loadPackage "CharacteristicClasses"; loadPackage "FMPIntersectionTheory";
i29 : X = last(P5=ZZ/16411[vars(0..5)],ideal(random(3,P5),random(3,P5),random(4,P5)));
i30 : (time Segre X,time SegreClass X,time segreClass X,time SegreClass(X,MathMode=>true));
-- used 0.1511 seconds
-- used 1.00936 seconds
-- used 34.1471 seconds
-- used 74.572 seconds
```

input	CC	Cremona (prob.)	FMPIntTh	Cremona (det.)
complete int. of type (3, 3, 4) in \mathbb{P}^5	0.15	1.01	34.15	74.57
rational normal surface $S(1, 4) \subset \mathbb{P}_{\mathbb{Q}}^6$	1.41	0.74	5.32	0.06
Grassmannian $\mathbb{G}(1, 4) \subset \mathbb{P}_{\mathbb{Q}}^9$	0.16	0.09	0.42	0.02
base locus of ϕ in Ex. 2.2	0.23	0.44	6.49	663.79
$X \subset \mathbb{P}^{11}$ in Ex. 2.3 over \mathbb{F}_{331^2}	65.76	83.66	–	0.85
$X \subset \mathbb{P}^{11}$ in Ex. 2.3 over $\mathbb{Z}/16411$	3.62	11.61	198.92	0.74

Table 2. Run-times to compute Segre classes in `CharacteristicClasses.m2`, `FMPIntersectionTheory.m2`, and `Cremona.m2` (all times given in seconds).

SUPPLEMENT. The [online supplement](#) contains version 4.2.2 of `Cremona.m2`.

REFERENCES.

- [Aluffi 2003] P. Aluffi, “Computing characteristic classes of projective schemes”, *J. Symbolic Comput.* **35**:1 (2003), 3–19. [MR](#) [Zbl](#)
- [Dolgachev 2011] I. Dolgachev, “Lectures on Cremona transformations”, 2011, available at <http://www.math.lsa.umich.edu/~idolga/cremonalect.pdf>.
- [Fulton 1984] W. Fulton, *Intersection theory*, Ergebnisse der Mathematik und ihrer Grenzgebiete (3) **2**, Springer, 1984. [MR](#) [Zbl](#)
- [Harris 1992] J. Harris, *Algebraic geometry, a first course*, Graduate Texts in Mathematics **133**, Springer, 1992. [MR](#) [Zbl](#)
- [Harris 2017] C. Harris, “Computing Segre classes in arbitrary projective varieties”, *J. Symbolic Comput.* **82** (2017), 26–37. [MR](#) [Zbl](#)
- [Hartshorne 1977] R. Hartshorne, *Algebraic geometry*, Graduate Texts in Mathematics **52**, Springer, 1977. [MR](#) [Zbl](#)
- [Helmer 2016] M. Helmer, “Algorithms to compute the topological Euler characteristic, Chern–Schwartz–MacPherson class and Segre class of projective varieties”, *J. Symbolic Comput.* **73** (2016), 120–138. [MR](#) [Zbl](#)
- [Jost 2015] C. Jost, “Computing characteristic classes and the topological Euler characteristic of complex projective schemes”, *J. Softw. Algebra Geom.* **7** (2015), 31–39. [MR](#)
- [Macaulay2] D. R. Grayson and M. E. Stillman, “Macaulay2: a software system for research in algebraic geometry”, available at <http://www.math.uiuc.edu/Macaulay2>.
- [Miller and Sturmfels 2005] E. Miller and B. Sturmfels, *Combinatorial commutative algebra*, Graduate Texts in Mathematics **227**, Springer, 2005. [MR](#) [Zbl](#)
- [Mumford 1988] D. Mumford, *The red book of varieties and schemes*, Lecture Notes in Mathematics **1358**, Springer, 1988. [MR](#) [Zbl](#)
- [Russo and Simis 2001] F. Russo and A. Simis, “On birational maps and Jacobian matrices”, *Compositio Math.* **126**:3 (2001), 335–358. [MR](#) [Zbl](#)
- [Russo and Staglianò 2017] F. Russo and G. Staglianò, “Congruences of 5-secant conics and the rationality of some admissible cubic fourfolds”, 2017. [arXiv](#)
- [Simis 2004] A. Simis, “Cremona transformations and some related algebras”, *J. Algebra* **280**:1 (2004), 162–179. [MR](#) [Zbl](#)
- [Staglianò 2016] G. Staglianò, “Examples of special quadratic birational transformations into complete intersections of quadrics”, *J. Symbolic Comput.* **74** (2016), 635–649. [MR](#) [Zbl](#)
- [Staglianò 2018] G. Staglianò, “Special cubic Cremona transformations of \mathbb{P}^6 and \mathbb{P}^7 ”, *Adv. Geom.* (online publication March 2018).

RECEIVED: 19 Jan 2017

REVISED: 24 May 2018

ACCEPTED: 11 Jun 2018

GIOVANNI STAGLIANÒ:

giovannistagliano@gmail.com

Dipartimento di Ingegneria Industriale e Scienze Matematiche, Università Politecnica delle Marche, Ancona, Italy

<i>HeLP: a GAP package for torsion units in integral group rings</i> Andreas Bächle and Leo Margolis	1
<i>A software package to compute automorphisms of graded algebras</i> Simon Keicher	11
<i>A package for computations with classical resultants</i> Giovanni Staglianò	21
<i>The SpaceCurves package in Macaulay2</i> Mengyuan Zhang	31
<i>The ReesAlgebra package in Macaulay2</i> David Eisenbud	49
<i>A Macaulay2 package for computations with rational maps</i> Giovanni Staglianò	61
<i>ExteriorIdeals: a package for computing monomial ideals in an exterior algebra</i> Luca Amata and Marilena Crupi	71
<i>Software for computing conformal block divisors on $\overline{M}_{0,n}$</i> David Swinarski	81
<i>Divisor Package for Macaulay2</i> Karl Schwede and Zhaoning Yang	87