FUNCTIONALITIES FOR GENUS 2 AND 3 CURVES

REYNALD LERCIER, CHRISTOPHE RITZENTHALER, AND JEROEN SIJSLING

Abstract. We gather and illustrate some functions that we wrote in the Magma computer algebra system for curves of genus 2 and 3. In genus 3, we furnish functions both for non-hyperelliptic and for hyperelliptic curves. A fair bit of the functionality in the latter case extends to hyperelliptic curves of arbitrary genus.

This is a technical note to illustrate new functions that we implemented to complete those already included in Magma 2.25-7. Most of them were disseminated in previous articles and we gather them here in clean packages for the users' convenience. The packages and the full description of the functionalities and options can be found at [LRS20a] for the hyperelliptic case and at [LRS20b] for quartics.

1. Hyperelliptic curves

In this section, we describe functionality for arbitrary genus hyperelliptic curves of genus $g > 1$ that are given by a smooth hyperelliptic model $y^2 = f(x)$ with $f \in k[x]$ of degree $2g+1$ or $2g+2$. We assume the characteristic p of the base field k to be different from 2.

1.1. Computation of isomorphisms.

Let $C_i : y^2 = f_i(x)$ be two hyperelliptic curves of genus g over a base field k. Isomorphisms $C_1 \to C_2$ are of the form

$$(x, y) \mapsto \left(\frac{ax+b}{cx+d}, \frac{ey}{(cx+d)^{2g+2}} \right) \tag{1.1}$$

with $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix} \right] \in \mathrm{GL}_2(k)$ and $e \in k^*$. (Over an algebraically closed field, one can impose $e = 1$, but we do not insist on this.) The set of isomorphisms is a principal homogeneous space over the group of automorphisms of either of the curves C_1 and C_2. Determining if C_1 and C_2 are isomorphic, and returning the set of isomorphisms if they are, boils down to computing the elements of $\mathrm{GL}_2(k)$ whose right action transforms f_1 into a multiple of f_2.

One possibility for determining these isomorphisms is by applying Gröbner bases after a formal coefficient comparison, see for example [Gö03]. In [LRS12], we gave alternative algorithms that speed up this computation when p does not divide $2g + 2$. Our function `IsIsomorphicHyperellipticCurves()` determines both the full set of isomorphisms $C_1 \to C_2$ over k itself and that over the algebraic closure of k using the option `geometric`. We refer to the latter as geometric isomorphisms. In this case the option `commonfield` can be enabled to return the list of isomorphisms embedded into a common overfield. When working over a number field, this can be an expensive operation. Using the option `covariant` (which is the default) performs the calculation of these isomorphisms using the reduction process involving covariants in [LRS12], whereas disabling this option uses the more direct method from *loc. cit.* Finally, concentrating on the matrix elements only leads to the definition of *reduced* isomorphisms or automorphisms, for which we have created dedicated functions as well (for instance `ReducedAutomorphismsOfHyperellipticCurve`).

Date: February 14, 2021.

2010 *Mathematics Subject Classification.* 14H10, 14H25, 14H37, 14H50.

Key words and phrases. plane quartic curves, hyperelliptic curves, invariants, reconstruction, descent, twists.

Example 1.1. We determine the isomorphisms $C_1 \to C_2$ for $C_1 : y^2 = x^{12} - 1$ and $C_2 : y^2 = x^{12} + 1$. The code and the resulting output are as follows.

```
> P<x> := PolynomialRing(Rationals());
> C1 := HyperellipticCurve(x^12 - 1);
> C2 := HyperellipticCurve(x^12 + 1);
> test, _ := IsIsomorphicHyperellipticCurves(C1, C2);
> test;
false
> test, _ := IsIsomorphicHyperellipticCurves(C1, C2 : geometric := true);
> test;
true
```

Isomorphism functions enable to determine the group of automorphisms of an hyperelliptic curve as well. Given a curve defined over a field of characteristic different from 2, the function `AutomorphismGroupOfHyperellipticCurve` returns a permutation group, followed, if the option `explicit` is enabled, by an isomorphism to the group of automorphisms of the curve over its base ring or its algebraic closure. As for the isomorphism functionality, these group calculations are more efficient than the generic Magma functionality.

Example 1.2. We determine automorphism group of the curve $C_1 : y^2 = x^{12} - 1$ over the finite field $\mathbb{F}_{11}$.

```
> P<x> := PolynomialRing(GF(11));
> C := HyperellipticCurve(x^12 - 1);
> aut, phi := AutomorphismGroupOfHyperellipticCurve(C : geometric := true,
explicit := true);
> aut;
Permutation group aut acting on a set of cardinality 2640
Order = 2640 = 2^4 * 3 * 5 * 11
> GroupName(aut);
C2.PSL(2,11).C2
> [phi(aut.i) : i in [1..Ngens(aut)]];
[
    <
        [      1      0]
        [      0      1],

        10
    >,

    <
        [      1  $.1^27]
        [ $.1^77  $.1^20],

        4
    >,
    <
        [      1  $.1^98]
        [ $.1^28  $.1^30],

        8
    >,

    <
        [      1 $.1^115]
        [  $.1^5      10],

        $.1^6
    >,

    <
        [      1  $.1^85]
        [ $.1^35      10],

        $.1^6
    >
]
```

1.2. **Twists.** The computation of representatives for all possible twists of a hyperelliptic curve $C : y^2 = f(x)$ over a finite field of characteristic different from 2 is implemented using [CN07], in particular to rule out so-called self-dual curves. It strongly relies on the computation of the geometric automorphism group of C. This is implemented by the function `Twists()`. If the option `AutomorphismGroup` is set `true`, it also outputs the group of reduced automorphisms (i.e. the subgroup of $\mathrm{PGL}_2(\bar{k})$ generated by the first part of the representation (1.1) of the geometric automorphisms of C) as an abstract permutation group. This may currently slow down the algorithm when the group is large, and here there remains room for improvement.

Example 1.3. We determine the twists of the hyperelliptic curve $C : y^2 = x^{12} - 1$ over the finite field $\mathbb{F}_{11}$.

```
> P<x> := PolynomialRing(GF(11));
> C := HyperellipticCurve(x^12 - 1);
> Ts, Aut := Twists(C : AutomorphismGroup := true);
> Ts;
[
    Hyperelliptic Curve defined by y^2 = x^12 + 10 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 10*x^11 + 5*x + 3 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 3*x^11 + 8*x + 8 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 6*x^11 + 3*x over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 5*x^11 + 7*x + 3 over GF(11),
    Hyperelliptic Curve defined by y^2 = 2*x^12 + 10*x^11 + 3*x + 6 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 7*x^11 + 5*x over GF(11),
    Hyperelliptic Curve defined by y^2 = x^11 + 10*x over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 3*x^11 + 7*x + 4 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 2*x^11 + 9*x + 5 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 4*x^11 + 9*x + 5 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 4*x^11 + 5*x + 4 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 4*x^11 + 2*x + 3 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^12 + 7*x^11 + 7*x + 9 over GF(11)
]
> Aut;
Permutation group Aut acting on a set of cardinality 12
Order = 1320 = 2^3 * 3 * 5 * 11
    (3, 4, 7, 5, 9)(6, 10, 8, 11, 12)
    (3, 6, 5, 11, 4, 10, 9, 12, 7, 8)
    (2, 3)(4, 5)(7, 12)(8, 10)(9, 11)
    (1, 2)(4, 9)(5, 7)(6, 8)(11, 12)
> GroupName(Aut);
PSL(2,11).C2
```

2. Hyperelliptic curves of genus 2 and 3

2.1. **Invariants.** Over an algebraically closed field k of characteristic p with $p \neq 2$, the isomorphism class of a genus g hyperelliptic curve $y^2 = f(x)$ with $f \in k[x]$ corresponds to the orbit of the binary form $z^{2g+2} f(x/z)$ under the classical action of $\mathrm{GL}_2(k)$. One can therefore (see [Dol03, §10.2]) characterize these classes by the invariant space $\mathrm{Proj}(R_g(k))$ with $R_g(k) = (\oplus_{n \geq 0} \mathrm{Sym}^n(\mathrm{Sym}^{2g+2}(k^2)))^{\mathrm{SL}_2(k)}$. Working out generators for the algebra $R_g(\mathbb{C})$ was a popular pastime of nineteenth-century mathematicians. For $g = 2$ (that is, for sextic binary forms), this determination goes back to [Cle72], whereas for $g = 3$ (that is, for octic binary forms) it goes back to [SF79, VG88].

When $p \neq 0$, the situation is more involved. In order to obtain a set of generators for $R_g(k)$, a good starting point is often to reduce a set of well-normalized generators of $R_g(\mathbb{C})$ modulo p. (Here well-normalized means to be primitive $\mathbb{Z}$-integral polynomials in the coefficients of a generic form; such a normalization is always possible by [Sil92, Lemma.5.8.1].) Unfortunately, there is currently no way to easily check whether this reduced set of generators will indeed generate $R_g(k)$. There are examples, for instance $g = 3$ and $p = 5$, where this is not the case.

For $g = 2$, Igusa [Igu60] managed to give a "universal set of invariants", which works in every characteristic, including 2. This set of invariants $\{I_2, I_4, I_6, I_8, I_{10}\}$ (with I_8 being superfluous except when the characteristic equals 2) is integrated in Magma and can be called by the function `IgusaInvariants()`[1].

For $g = 3$, thanks to the work of [Gey74], one could show in [LR12][2] that the reduction of Shioda generators for $R_3(\mathbb{C})$ are still generators for $R_3(k)$ when the characteristic of k is greater than 7. For all smaller characteristics save $p = 5$ but including 2, [Bas15] was able to give a set of separating invariants (i.e., invariants that allow one to separate the orbits of the binary form and therefore to characterize the isomorphism classes). We conjecture that they are also generators.

Example 2.1. The relevant function for invariants of hyperelliptic curves in genus 3 is `ShiodaInvariants`. It returns a list $\underline{I}$ of elements of k to be considered as an element in a weighted projective space with

[1] There are also other sets of invariants (IgusaClebschInvariants(), ClebschInvariants()) and absolute invariants (`G2Invariants()`), which are used for historical or practical reasons.

[2] Said reference should have included a different proof of [Shi67, Lemma 1], as the one in *loc. cit.* is only valid in characteristic 0. A general proof can be found in [Smi95, Prop.5.5.2].

weights which are indicated by the second output of the function. This list depends on the characteristic p (and when $p = 2$ on the type defined in [NS04] or [Bas15, Appendix]). When $p > 7$, it is a list of generators for $R_3(k)$ of weight $(2, 3, \dots, 10)$. For $p = 2, 3, 7$, it is a list of separating invariants and for $p = 5$ a minimal list of invariants that generates the largest subring of invariants that we have been able to determine so far. Note that the function has a flag `PrimaryOnly` which only outputs a (proven) homogeneous system of parameters in all characteristic different from 2.

Two lists $\underline{I}$ can be normalized and compared (and therefore the corresponding curves seen to be isomorphic or not) using the flag `normalize:=true`, which relies on the techniques of [LR12, Sec.1.4]. Another flag is `IntegralNormalization`, which multiplies the Shioda invariants by certain constants so that the invariants are defined over $\mathbb{Z}$. One can also choose to get only part of the list of invariants by filtering them using `degmin,degmax`.

```
> P<x> := PolynomialRing(GF(3));
> ShiodaInvariants(x^8 + 1);
[ 1, 0, 0, 0, 0, 0, 1, 0, 1, 2 ]
[ 2, 3, 4, 5, 6, 7, 8, 9, 10, 12 ]
```

2.2. **Reconstruction from invariants.** Given the values $\underline{I}$ of a set of generators for the invariants of a hyperelliptic curve $C : y^2 = f(x)$ over a field k, reconstructing from the invariants means being able to produce, given only $\underline{I}$, a model $D : y^2 = g(x)$ that is $\bar{k}$-isomorphic to C. One may try to find D over the ground field k itself, and if possible with "small coefficients" when for example $k = \mathbb{Q}$.

The general philosophy to reconstruct hyperelliptic curves from the knowledge of their invariants is explained in [Mes91] and worked out there for a generic genus 2 hyperelliptic curve. It starts with three covariants of order 2 and then uses beautiful formulas due to Clebsch [Cle72, §103] to construct a conic and a plane curve of degree $g + 1$ whose intersection are the Weierstrass points of C. In order to find a hyperelliptic model over k, one needs to find a k-rational point for the conic. In the sequel, we will call fields for which Magma can do this computable fields. At any rate, it is easier to produce a model over a quadratic extension of k. For $g = 2$, the general case over any computable field (also in characteristic 2) was implemented in Magma using the work of [CQ05, CNP05]. The corresponding function is called `HyperellipticCurveFromIgusaInvariants()`.

In [LR12], this functionality was extended to hyperelliptic curves of genus 3 for computable fields of characteristic $p > 7$. Now, thanks to Basson's work [Bas15], we can also reconstruct over any computable field of characteristic different from 5.

Note that the reconstruction functions also return the geometric automorphism group of the curve as an abstract permutation group. This involves only the invariants of the curve, as known relations between these describe the locus of curves with a given geometric automorphism group. Also note that there exist a function `HyperellipticPolynomialFromIgusaInvariants()` (resp. a function `HyperellipticPolynomialFromShiodaInvariants()`) that can be applied to the invariants of any GIT-stable sextic (resp. octic binary form). Recall that such a form is defined by having no factor of multiplicity greater or equal to 4.

Example 2.2. We reconstruct a curve from invariants over the base field $\mathbb{F}_3$.

```
> I := [ GF(3)!1, 0, 0, 0, 0, 0, 1, 0, 1, 2 ];
> HyperellipticCurveFromShiodaInvariants(I);
```

```
Hyperelliptic Curve defined by y^2 = x^8 + 2 over GF(3)
Permutation group acting on a set of cardinality 32
    (1, 19, 3, 17)(2, 20, 4, 18)(5, 23, 7, 21)(6, 24, 8, 22)(9, 27, 11, 25)(10,
        28, 12, 26)(13, 31, 15, 29)(14, 32, 16, 30)
    (1, 9)(2, 10)(3, 11)(4, 12)(5, 13)(6, 14)(7, 15)(8, 16)(17, 29)(18, 30)(19,
        31)(20, 32)(21, 26)(22, 25)(23, 28)(24, 27)
```

Over $\mathbb{Q}$, we optimized the reconstruction to get models with small integer coefficients. This was already implemented for $g = 3$ in [KLL+18, Sec.3.1] using a tricks involving a so-called "variation of conics" which fastens the search phase for a rational point on the conic involved in the reconstruction. Oddly a similar trick was harder to develop for $g = 2$. The algebra $R_2(\mathbb{C})$ is generated by the invariants I_2, I_4, I_6, I_{10} which are algebraically independent and with I_{15} such that $I_{15}^2 \in \mathbb{Z}[I_2, I_4, I_6, I_{10}]$. The latter is useful to classify sextic forms under the action of $\mathrm{SL}_2(\mathbb{C})$, yet it becomes irrelevant in the context of classifying curves of genus 2. The reason for this is that the corresponding Proj remains identical when considering only the sub-algebra of elements of even degree, that is, sub-algebra $\mathbb{Z}[I_2, I_4, I_6, I_{10}]$. Therefore, when computing the invariants with `IgusaInvariants()`, I_{15} is not stored unless the flag `extend` is set to `true`.

We could not find a system of four covariants of order 2 to perform variation of conic method which would not involve I_{15} in the reconstruction process. Fortunately, if I_{15} is not provided, there exists a relation of degree 30 that gives I_{15}^2 as a function of I_2, I_4, I_6 and I_{10}. And if by misfortune this relation is not a square over $\mathbb{Q}$, we can substitute $\lambda\, I_2$, $\lambda^2\, I_4$, $\lambda^3\, I_6$ and $\lambda^5\, I_{10}$ for I_2, I_4, I_6, $\ldots I_{10}$ where λ is a constant chosen such that $\lambda^{15}\, I_{15}^2$ is now a square. This yields I_{15} up to a sign, which suffices for our purposes. (Note that the genus two curves $y^2 = f(x)$ and $y^2 = -f(x)$ are twists and have the same even degree Igusa invariants and $I_{15}(f) = -I_{15}(-f)$.)

At the end of the procedure, the function `MinRedBinaryForm` is used to get even smaller coefficients.

Example 2.3. We reconstruct a curve of genus 3 from its invariants.

```
> P<x> := PolynomialRing(Rationals());
> f := x^8 + x^3 + 1;
> f2 := P ! (Evaluate(f, (x+1001)/(3*x+5))*(3*x+5)^8);
> f2;
6805*x^8 + 827242*x^7 + 765112882*x^6 + 306007035874*x^5 + 72331829214220*x^4 +
    5628736468095l806*x^3 + 2816843187239852927 8*x^2 + 805616828969271682 4758*x
    + 100802805607319041277 6751
> I := ShiodaInvariants(f2);
> HyperellipticCurveFromShiodaInvariants(I);
Hyperelliptic Curve defined by y^2 = x^8 + x^3 + 1 over Rational Field
Symmetric group acting on a set of cardinality 2
Order = 2
```

2.3. **Twists over finite fields.** For a genus 2 curve over a finite field (including characteristic 2), the computation of twists has been implemented in all cases. This is now extended similarly to genus 3 using a case-by-case study to work out the explicit coboundaries for specific models determined by the invariants of the initial curve when the automorphism group is large (or in characteristic 2) and by the generic method of Section 1.2 for small automorphism groups.

Example 2.4. We determine the twists of the hyperelliptic curve $C : y^2 = x^8 + 1$ over $\mathbb{F}_{11}$.

```
> P<x> := PolynomialRing(GF(11));
> C := HyperellipticCurve(x^8 + 1);
> Twists(C);
[
    Hyperelliptic Curve defined by y^2 = x^8 + 10 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^8 + 5*x^7 + 4*x^6 + 3*x^5 + 7*x^4 +
        x^2 + 3*x + 2 over GF(11),
    Hyperelliptic Curve defined by y^2 = 2*x^8 + 10*x^7 + 8*x^6 + 6*x^5 + 3*x^4
        + 2*x^2 + 6*x + 4 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^7 + 10*x^6 + 2*x^4 + 3*x^3 + 5*x + 1
    over GF(11),
    Hyperelliptic Curve defined by y^2 = x^8 + 2*x^7 + 4*x^6 + x^5 + 10*x^4 +
        2*x^3 + 10*x^2 + 9*x + 9 over GF(11),
    Hyperelliptic Curve defined by y^2 = 2*x^8 + 4*x^7 + 8*x^6 + 2*x^5 + 9*x^4 +
        4*x^3 + 9*x^2 + 7*x + 7 over GF(11),
    Hyperelliptic Curve defined by y^2 = x^8 + 3 over GF(11),
    Hyperelliptic Curve defined by y^2 = 2*x^8 + 6 over GF(11)
]
```

3. Plane quartics

3.1. **Dixmier–Ohno invariants.** Isomorphisms of plane smooth quartics over an algebraically closed field k are induced by linear transformations of the ambient projective plane $\mathbb{P}^2$. Therefore, isomorphism classes are characterized by the space $\mathrm{Proj}(R(k))$ where $R(k) = (\oplus_{n\geq 0} \mathrm{Sym}^n(\mathrm{Sym}^4(k^3))^{\mathrm{SL}_3(k)}$, i.e. the ring of invariants of quartic ternary forms under the classical action of $\mathrm{SL}_3(k)$. When k is of characteristic 0, Dixmier [Dix87] gave a list of 7 invariants which form a homogeneous system of parameters. It was completed by [Ohn07], who furnished a list of 13 generators for the algebra $R(\mathbb{C})$. This invariants are polynomials in the 15 coefficients of a ternary quartic forms with coefficients in $\mathbb{Z}[1/6]$. They can be considered as a point in the weighted projective space with weights $(3, 6, 9, 9, 12, 12, 15, 15, 18, 18, 21, 21, 27)$. Corresponding functionality was implemented in Magma for the first time in [GK06]. The current function is called `DixmierOhnoInvariants()`, and differs from the previous implementations up to some normalization constants. To recover the initial implementation in [GK06], it suffices to set the flag `IntegralNormalization` equal to `true`. Normalized representatives (for which it suffices to test for equality to decide whether the curves involved are geometrically isomorphic) can also be obtained, namely by setting the flag `normalize` to `true`.

Example 3.1. We consider the Klein quartic and one of its non-trivial twists over $\mathbb{Q}$.

```
> P<x,y,z> := PolynomialRing(Rationals(), 3);
> PP := ProjectiveSpace(P);
> f1 := x^3*y + y^3*z + z^3*x;
> f2 := x^4 + 7*x^3*z + 3*x^2*y^2 - 3*x^2*z^2 - 6*x*y*z^2 - 5*x*z^3 +
> 2*y^3*z + 3*y^2*z^2 + 2*y*z^3 - 4*z^4;
> C1 := Curve(PP, f1); DO1 := DixmierOhnoInvariants(C1 : normalize := true);
> C2 := Curve(PP, f2); DO2 := DixmierOhnoInvariants(C2 : normalize := true);
> DO1 eq DO2;
true
> IsIsomorphicPlaneQuartics(C1, C2);
false []
```

A list of generators of the invariants of smooth plane quartics in positive characteristics is not known, although it is suspected that the reduction of the Dixmier–Ohno invariants are generators when the characteristic is greater than 7. In [LLGR20] homogeneous systems of parameters are determined in all characteristics except 3, for which there is a conjectural HSOP that involves an invariant of degree 81. A call to `DixmierOhnoInvariants()` in general characteristic outputs a minimal set of invariants that generate the largest subring of invariants that we know of so far.

Among the Dixmier–Ohno invariants of a form $f(x, y, z)$, the invariant I_{27} of degree 27 plays a particular role. It can be shown that $\frac{1}{2^{40}} I_{27}$ has integral coefficients and that over any field, its zero locus is precisely the locus of singular plane quartics. The construction in [GK06] computes the resultant of the 3 partial derivatives of f after [GKZ94, p.426]. Unfortunately, this method fails in characteristic 2 for intrinsic reasons, and the original implementation also had issues in characteristic 3. This is why in these characteristics, we now compute the resultant of the partial derivatives of f with respect to two of the variables x and y, and of the form f itself. This follows an idea and used programs kindly provided to us by Laurent Busé, which are based on the techniques developed in [BJ14, Def.4.6, Prop.4.7] or [Dem12, Prop.11] and [Jou97, Sec.3.11.19.25]. We obtain a parasitical factor which is the discriminant of the binary form $f_{|z=0}$. However, one can get rid of this issue by deforming the form f into $f + \varepsilon(x^4 + y^4)$ in characteristic different from 2 and $f + \varepsilon(x^4 + xy^3 + y^4)$ in characteristic 2. It then suffices to compute the discriminant of the family thus obtained and to take its value for $\varepsilon = 0$.

Example 3.2. We compute the discriminant of the Klein Quartic over $\mathbb{F}_2$.

```
> P<x,y,z> := PolynomialRing(GF(2), 3);
> Q := x^3*y + y^3*z + z^3*x;
> DiscriminantOfTernaryQuartic(Q);
1
```

3.2. **Reconstruction of quartics.** Given the Dixmier–Ohno invariants $\underline{I}$ of a generic plane smooth quartic C over a computable field k of characteristic 0, the algorithms developed in [LRS18] allow the reconstruction of a model of this quartic, which is returned over k itself as long as the geometric automorphism group of C is not of order 2. The relevant function is `PlaneQuarticFromDixmierOhnoInvariants(I)`. We remark the following:

(i) In the above, "generic" means concretely that the invariant I_{12} is different from 0. If I_{12} is zero, other systems of co- or contra-variants may be chosen to perform the reconstruction. These variants have not been implemented, and there are smooth plane quartics, like the Klein quartic, for which no such system exists. Regardless, for all non-trivial automorphism strata except for the cyclic group $\mathbb{Z}/2\mathbb{Z}$, as well as for $(\mathbb{Z}/2\mathbb{Z})^2$ in case $I_{12} = 0$, an *ad hoc* reconstruction is performed.

(ii) If one would know that the Dixmier–Ohno invariants are generators of $R(k)$, then reconstruction from invariants is possible, at least when the characteristic p of k is large enough. Currently, it is not clear when this is the case and the best is to try if the algorithm returns a result (which is then correct). For now we remark that the primes $p \leq 13$ or $p = 79$ are problematic for the generic stratum, and that primes up to 41762629 can be problematic for curves with non-trivial automorphism group.

(iii) If the quartic curve has automorphism group of order 2, the field of moduli is not necessarily a field of definition and the reconstruction may happen over a quadratic extension only. Still, the algorithms will in practice often find a model over the field of moduli if it exists.

(iv) When $k = \mathbb{Q}$, the variation of conics and the algorithms of [Els09] yield a reconstruction of quartics with small coefficients, as in [KLL$^+$18].

Example 3.3. We reconstruct a plane quartic from its invariants.

```
> P<x,y,z> := PolynomialRing(GF(31), 3);
> PP := ProjectiveSpace(P);
> f1 := x^4 + 3*y^4 + 5*z^4 + x^2*y*z + x*y*z^2 + x^2*y^2;
> C1 := Curve(PP, f1);
> I := DixmierOhnoInvariants(f1);
> C2 := Curve(PP, TernaryQuarticFromDixmierOhnoInvariants(I));
> IsIsomorphicPlaneQuartics(C1, C2);
true [
    [ 1 24  8]
    [ 8 27 20]
    [13 20 19]
]
```

3.3. **Isomorphisms.** Isomorphisms and automorphisms of plane quartics have been implemented following the covariant method due to van Rijnswou [vR01] that is also used in another form in the reconstruction algorithms. Let C_1 and C_2 be two plane quartic curves over a field k. Our algorithm first checks for equality of normalized Dixmier–Ohno invariants of C_1 and C_2, since if this equality does not hold, no isomorphisms can exist.

If this condition is satisfied, the algorithms first try to find the actual isomorphisms $C_1 \to C_2$ under the assumption that $I_{12} \neq 0$. In this case, [vR01] shows that the use of a suitable covariant reduces this question to finding transformations between certain binary forms associated to C_1 and C_2, which leads us to the same computation of elements in $\mathrm{GL}_2(k)$ that was considered in Section 1.1. In non-generic cases, we have used a direct Gröbner basis method due to Michael Stoll (private communication).

Once again the algorithms admit both a version over the base field and a geometric version, with the latter finding the isomorphisms over the algebraic closure of k. Both versions are very efficient over finite fields, and the version over the base field is also reasonably fast for $k = \mathbb{Q}$. By contrast, finding geometric isomorphisms between plane quartic curves over the rationals can still take a fair amount of time. For more general fields, the implementation still takes too long, and our functions therefore restrict considerations to the cases where k is either finite or the rational field.

Example 3.4. We determine the automorphisms of a plane quartic over the rationals.

```
> P<x,y,z> := PolynomialRing(Rationals(), 3);
> PP := ProjectiveSpace(P);
> C := Curve(PP, x^3*y+y^3*z+z^3*x);
> aut, phi := AutomorphismGroupOfPlaneQuartic(C : geometric:=true, explicit :=
true);
> aut;
Permutation group aut acting on a set of cardinality 8
Order = 168 = 2^3 * 3 * 7
    (2, 3, 4)(5, 8, 7)
    (2, 4, 5)(3, 6, 7)
    (1, 2)(3, 7)(4, 5)(6, 8)
> GroupName(aut);
PSL(2,7)
> [phi(aut.i) : i in [1..Ngens(aut)]];
[
    [0 0 1]
    [1 0 0]
    [0 1 0],

    [0 0 -r1^5 - r1^4 - r1^3 - r1^2 - r1 - 1]
    [1 0 0]
    [0 r1^4 0],

    [1 r1^2 + r1 -r1^5 - r1^4 - r1^3]
    [-r1^4 - r1^3 - r1^2 - r1 - 1 r1^5 + r1^4 + r1^3 + r1^2 -r1^5 - r1^4 - r1^3
        - r1^2 - r1 - 1]
    [-r1^4 - r1^3 - r1^2 r1 r1^4 + r1^3]
]
```

Note that some of these routines may overlap with routines naively included in Magma. As in the hyperelliptic case, they are generally much faster.

Example 3.5. We compare timings for our automorphism routine and the native one included in Magma.

```
> P<x,y,z> := ProjectiveSpace(Rationals(),2);
> C := Curve(P, x^4+y^4+z^4);
> time aut, phi := AutomorphismGroupOfPlaneQuartic(C : geometric:=true, explicit
:= true);
Time: 0.880
> GroupName(aut);
C4^2:C3:C2
> // to get all automorphisms, we base change to Q(zeta_8)
> K := CyclotomicField(8);
> C1 := BaseChange(C,K);
> time G := AutomorphismGroup(C1);
Time: 3.090
> Gp,rep := MatrixRepresentation(G);
> GroupName(Gp);
C4^2:C3:C2
```

3.4. **Twists.** Using classical reductions to compute the cohomology set $H^1(\mathrm{Gal}(\bar{k}/k), \mathrm{Aut}(C))$ over finite fields (see for instance [MT10]), we give a function `Twists()` to compute a list of representatives of all twists of a smooth plane quartic over a finite field. This relies on the prior computation of the geometric automorphism group of C. Note that it relies on the function `Twists(C, H)` that takes as its input any quasi-projective curve C (not necessarily plane or non-singular) and any finite subgroup H of the geometric automorphism group C, and that computes the corresponding twists as long as the elements of H acts as linear transformations of the ambient space, which is for example the case when C is canonically embedded or smooth.

Example 3.6. We compute the twists of the Klein quartic over $\mathbb{F}_{31}$.

```
> P<x,y,z> := PolynomialRing(GF(31), 3);
> PP := ProjectiveSpace(P);
> f := x^3*y + y^3*z + z^3*x;
> C := Curve(ProjectiveSpace(P), f);
> #Twists(C);
4
```

4. Remaining work

For the benefit of the motivated reader, this section lists unanswered questions or functions that remain to be implemented. The number of stars reflects our naive estimation of the difficulty and/or quantity of the work involved.

- $\star$ Currently, a generic Magma functions determines the structure of the reduced automorphism group from the list of reduced automorphisms. This stands to be improved, using the classification of reduced automorphism groups. The answer to this question is easier if one is only interested in the abstract group structure, and more complicated if one also wishes to determine a map from this abstract group to the list of reduced automorphisms.
- $\star$ Find the separants for the invariant ring of binary octic forms in characteristic 5.
- $\star\star$ Prove that the separants for the invariant ring of binary octic forms in characteristic 3 and 7 (and 5?) are generators.
- $\star\star$ Reconstruct genus 3 hyperelliptic curves from a list of invariants (or separants) in characteristic 5.
- $\star\star$ Prove the correctness of the conjectural HSOP in characteristic 3.
- $\star\star$ Prove that the reductions of the Dixmier-Ohno invariants are still generators for the invariant ring if the characteristic of the residue field is larger than 7.
- $\star\star\star$ Determine generators for the ring of invariants in smaller characteristic.
- $\star\star\star$ Make the reconstruction process for generic plane quartics work for all characteristics (or at least for those greater than 7).
- $\star\star\star\star$ The same question as the previous one, but this time for all quartics.

References

[Bas15] R. Basson. *Arithmétique des espaces de modules des courbes hyperelliptiques de genre* 3 *en caractéristique positive*. PhD thesis, Université de Rennes 1, Rennes, 2015.

[BJ14] L. Busé and J.-P. Jouanolou. On the discriminant scheme of homogeneous polynomials. *Math. Comput. Sci.*, 8(2):175–234, 2014.

[Cle72] A. Clebsch. *Theorie der binären algebraischen formen*. Verlag von B.G. Teubner, Leipzig, 1872.

[CN07] G. Cardona and E. Nart. Zeta function and cryptographic exponent of supersingular curves of genus 2. In *Pairing-based cryptography—Pairing 2007*, volume 4575 of *Lecture Notes in Comput. Sci.*, pages 132–151. Springer, Berlin, 2007.

[CNP05] G. Cardona, E. Nart, and J. Pujolàs. Curves of genus two over fields of even characteristic. *Math. Zeitschrift*, 250:177–201, 2005.

[CQ05] G. Cardona and J. Quer. Field of moduli and field of definition for curves of genus 2. In *Computational aspects of algebraic curves*, volume 13 of *Lecture Notes Ser. Comput.*, pages 71–83, Hackensack, NJ,, 2005. World Sci. Publ.

[Dem12] M. Demazure. Résultant, discriminant. *Enseign. Math. (2)*, 58(3-4):333–373, 2012.

[Dix87] J. Dixmier. On the projective invariants of quartic plane curves. *Adv. in Math.*, 64:279–304, 1987.

[Dol03] I. Dolgachev. *Lectures on invariant theory*, volume 296 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2003.

[Els09] A.-S. Elsenhans. Good models for cubic surfaces. Preprint at https://math.uni-paderborn.de/fileadmin/-mathematik/AG-Computeralgebra/Preprints-elsenhans/red_5.pdf, 2009.

[Gey74] W. D. Geyer. Invarianten binärer Formen. In *Classification of algebraic varieties and compact complex manifolds*, pages 36–69. Lecture Notes in Math., Vol. 412. Springer, Berlin, 1974.

[GK06] M. Girard and D. R. Kohel. Classification of genus 3 curves in special strata of the moduli space. In Hess, F. (ed.) et al., Algorithmic number theory. 7th international symposium, ANTS-VII, Berlin, Germany, July 23–28, 2006. Proceedings. Berlin: Springer. Lecture Notes in Computer Science 4076, 346-360 (2006)., 2006.

[GKZ94] I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, resultants, and multidimensional determinants*. Mathematics: Theory & Applications. Birkhäuser Boston Inc., Boston, MA, 1994.

[Gö03] N. Göb. Computing the automorphism groups of hyperelliptic function fields, 2003.

[Igu60] J.-I. Igusa. Arithmetic variety of moduli for genus two. *Ann. Math*, 72:612–649, 1960.

[Jou97] J. P. Jouanolou. Formes d'inertie et résultant: un formulaire. *Adv. Math.*, 126(2):119–250, 1997.

[KLL$^+$18] P. Kılıçer, H. Labrande, R. Lercier, C. Ritzenthaler, J. Sijsling, and M. Streng. Plane quartics over $\mathbb{Q}$ with complex multiplication. *Acta Arith.*, 185(2):127–156, 2018.

[LLGR20] R. Lercier, Q. Liu, E. L. García, and C. Ritzenthaler. Reduction type of smooth quartics, 2020.

[LR12] R. Lercier and C. Ritzenthaler. Hyperelliptic curves and their invariants: geometric, arithmetic and algorithmic aspects. *J. Algebra*, 372:595–636, 2012.

[LRS12] R. Lercier, C. Ritzenthaler, and J. Sijsling. Fast computation of isomorphisms of hyperelliptic curves and explicit descent. In E. W. Howe and K. S. Kedlaya, editors, *Proceedings of the Tenth Algorithmic Number Theory Symposium*, pages 463–486. Mathematical Sciences Publishers, 2012.

[LRS18] R. Lercier, C. Ritzenthaler, and J. Sijsling. Reconstructing plane quartics from their invariants. *Discrete & Computational Geometry*, pages 1–41, 2018.

[LRS20a] R. Lercier, C. Ritzenthaler, and J. Sijsling. `hyperelliptic`, a `Magma` repository for reconstruction and isomorphisms of hyperelliptic curves. https://github.com/JRSijsling/hyperelliptic, 2020.

[LRS20b] R. Lercier, C. Ritzenthaler, and J. Sijsling. `quartic`, a Magma package for calculating with smooth plane quartic curves. https://github.com/JRSijsling/quartic, 2020.

[Mes91] J.-F. Mestre. Construction de courbes de genre 2 à partir de leurs modules. In *Effective methods in algebraic geometry*, volume 94 of *Prog. Math.*, pages 313–334, Boston, 1991. Birkäuser.

[MT10] S. Meagher and J. Top. Twists of genus three curves over finite fields. *Finite Fields Appl.*, 16(5):347–368, 2010.

[NS04] E. Nart and D. Sadornil. Hyperelliptic curves of genus three over finite fields of characteristic two. *Finite Fields and Their Applications*, 10:198–220, 2004.

[Ohn07] T. Ohno. The graded ring of invariants of ternary quartics I, 2007. Unpublished.

[SF79] J. J. Sylvester and F. Franklin. Tables of the Generating Functions and Groundforms for the Binary Quantics of the First Ten Orders. *Amer. J. Math.*, 2(3):223–251, 1879.

[Shi67] T. Shioda. On the graded ring of invariants of binary octavics. *American J. of Math.*, 89(4):1022–1046, 1967.

[Sil92] J. H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1992. Corrected reprint of the 1986 original.

[Smi95] L. Smith. *Polynomial invariants of finite groups*, volume 6 of *Research Notes in Mathematics*. A K Peters, Ltd., Wellesley, MA, 1995.

[VG88] F. Von Gall. Das vollständige Formensystem der binären Form 7ter Ordnung. *Math. Ann.*, 31:318–336, 1888.

[vR01] S. M. van Rijnswou. *Testing the equivalence of planar curves.* PhD thesis, Technische Universiteit Eindhoven, Eindhoven, 2001.

Reynald Lercier, DGA & Univ Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France.
Email address: reynald.lercier@m4x.org

Christophe Ritzenthaler, Univ Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France.
Email address: christophe.ritzenthaler@univ-rennes1.fr

Jeroen Sijsling, Institut für Algebra und Zahlentheorie, Universität Ulm, Helmholtzstrasse 18 D-89081 Ulm, Germany
Email address: jeroen.sijsling@uni-ulm.de