Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)

Software Documentation – An Efficient Surface Tool Box –

T. Beck

RICAM-Report 2008-08

Software Documentation

Tobias Beck*
RICAM, Austrian Academy of Sciences
Altenberger Straße 69, A–4020 Linz, Austria

February 14, 2008

We document software written by the author during its time at RICAM. Implementations are done in MAGMA [4] and provided for download [1]. They have been improved during a stay with the MAGMA-group at Sydney University. We use functionality for Diophantine equation solving by Jana Pílniková [5].

Contents

*We are grateful to the MAGMA-group at the University of Sydney for the warm welcome and for giving us the opportunity to improve the software in this exquisite environment.

1 Auxiliary Routines

In the beginning we describe a few auxiliary routines that have been used to implement the main algorithms but are not specific to them.

1.1 Solving Zero-Dimensional Systems

The first routines (found in `solve_system.mag`) are for geometrically solving zero-dimensional polynomial systems, *i.e.*, for computing points on the corresponding variety.

Function 1 (`AllRoots`)

Input:	`p : RngUPolElt`
Parameters:	`ExtName := "alpha", ExtCount := 0`
Output:	`List, RngElt`

Description of function: Find all roots to a polynomial $\mathtt{p} \in \mathbb{E}[x]$ (where $\mathbb{E}$ is a number field or the function field of a curve in characteristic zero) up to conjugacy over the ground field. *I.e.*, for each irreducible factor $f \mid \mathtt{p}$ return *exactly one* root in an extension isomorphic to $\mathbb{E}[x]/\langle f \rangle$. The first return value is a list with elements of the form (x, m) where x is a root of multiplicity m.

If the ground field has to be extended, the algebraic elements will be displayed as `ExtName_i` where `i` starts from `ExtCount`. The last return value is the value of `ExtCount` plus the number of field extensions that have been introduced. ◇

```
> Attach("solve_system.mag");
> Q := Rationals();
> S<t> := PolynomialRing(Q);
> f := 4*t^4 + 12*t^3 + 13*t^2 + 12*t + 9;
> AllRoots(f : ExtName := "beta");
[*
  [* -3/2, 2 *],
  [* beta_0, 1 *]
*] 1
```

The polynomial $f = 4t^4 + 12t^3 + 13t^2 + 12t + 9 = (t^2 + 1)(2t + 3)^2 \in \mathbb{Q}[t]$ has the two-fold root $-\frac{3}{2}$ and the simple roots $\pm i$ which are conjugate, hence, only one of them is returned and is given the name `beta_0`. This is the only field extension, so an extension count of one is returned.

Function 2 (`SolveZeroDimIdeal`)

Input:	`id : RngMPol`
Parameters:	`ExtName := "alpha", ExtCount := 0`
Output:	`List, RngIntElt`

Description of function: Find all solutions of the (at most) zero-dimensional ideal `id` of a multivariate polynomial ring $\mathbb{E}[x_1, \ldots, x_n]$ (where $\mathbb{E}$ is a number

field or the function field of a curve in characteristic zero) up to conjugacy over the ground field. *I.e.*, for each maximal ideal $\mathfrak{m} \supseteq$ `id` return *exactly one* root in an extension isomorphic to $\mathbb{E}[x_1, \ldots, x_n]/\mathfrak{m}$.

The first return value is the list of roots $((x_1, \ldots, x_n), m, d)$ where $(x_1, \ldots, x_n)$ is a solution of `id`, m its multiplicity and d the degree of the residue field over the ground field.

If the ground field has to be extended, the algebraic elements will be displayed as `ExtName_i` where `i` starts from `ExtCount`. The last return value is the value of `ExtCount` plus the number of field extensions that have been introduced. ◇

We go on with the above example:

```
> R<x,y> := PolynomialRing(Q,2);
> I := ideal<R | x*y, (y-1)-4*x^2>;
> SolveZeroDimIdeal(I : ExtName := "beta", ExtCount := 1);
[*
    <[ 0, 1 ], 1, 1>,
    <[ beta_1, 0 ], 1, 2>
*] 2
```

The ideal vanishes on the intersection of a translated parabola with the coordinate axes. The two intersections with $y = 0$ are again $\pm\frac{1}{2}i$. Because of our first call we have set `ExtCount := 1` and got a new unique name `beta_1`. For the next call we will use `ExtCount := 2`.

Function 3 (`SolveInProductSpace`)

Input:	`id : RngMPol`
Parameters:	`seq := [0], ExtName := "alpha", ExtCount := 0`
Output:	`List, RngIntElt`

Description of function: Find the finitely many roots of a system of equations in some product space. Here `id` is an ideal of a polynomial ring $\mathbb{E}[x_1, \ldots, x_n]$ and `seq` is an ordered sequence without multiples in $\{0, \ldots, n\}$ starting with 0. Say, `seq` has $r + 1$ elements. Then the `seq[i]` mark blocks of projective coordinates, *i.e.*, we consider `id` an ideal defining a finite set of points in

$$\left(\prod_{1 \le i \le r} \mathbb{P}^{\texttt{seq[i+1]}-\texttt{seq[i]}-1}_{\mathbb{E}} \right) \times \mathbb{E}^{n-\texttt{seq[r+1]}}.$$

Then `id` must be an r-dimensional ideal which is homogeneous with respect to the total degree in $x_{\texttt{seq[i]}+1}, \ldots, x_{\texttt{seq[i+1]}}$ for all $1 \le i \le r$.

For example, if $n = 7$ and `seq` $= [0, 2, 5]$ then $(x_1 : x_2)$ and $(x_3 : x_4 : x_5)$ are the blocks of projective coordinates and (x_6, x_7) are affine coordinates. Then `id` should be a two-dimensional ideal homogeneous w.r.t. total degrees for the first two blocks separately. It defines a point set in $\mathbb{P}^1_{\mathbb{E}} \times \mathbb{P}^2_{\mathbb{E}} \times \mathbb{E}^2$. If no projective coordinates are intended stick to the default value `seq` $= [0]$.

If the ground field has to be extended, the algebraic elements will be displayed as `ExtName_i` where `i` starts from `ExtCount`. The last return value is the value of `ExtCount` plus the number of field extensions that have been introduced. ◇

```
> P<x,y,z,w> := PolynomialRing(Q,4);
> I := ideal<P | y-z, x^2-y*z, w^2-4>;
> SolveInProductSpace(I : seq := [0,3], ExtName := "beta",
  ExtCount := 2);
[*
    [ 1, 1, 1, 2 ],
    [ -1, 1, 1, 2 ],
    [ 1, 1, 1, -2 ],
    [ -1, 1, 1, -2 ]
*] 1
```

This example says that, *e.g.*, the point $(1:1:1\ ;\ 2)$ is in the zero set of `I` w.r.t. $\mathbb{P}^3_{\mathbb{Q}} \times \mathbb{Q}$. *I.e.*, the ideal vanishes for all values $x = y = z$ and $w = 2$.

1.2 Linear Relations and Field Extensions

Let V be a finite $\mathbb{E}$-vector space and $x_1, \ldots, x_n$ a basis. Let further $\mathbb{F}$ be a finitely generated field extension of $\mathbb{E}$ (not necessarily algebraic). Then we have an inclusion $V \subseteq V' := V \otimes_{\mathbb{E}} \mathbb{F}$ and $x_1 \otimes 1, \ldots, x_n \otimes 1$ is a basis of V', and by abuse of language we write again $x_1, \ldots, x_n$. Let $W' \subset V'$ be a linear subspace defined by linear relations $\rho_i = (\rho_{i,1}, \ldots, \rho_{i,n})$, *i.e.*, $\rho_{i,1}c_1 + \cdots + \rho_{i,n}c_n = 0$ if and only if $c \in W'$. In other words W', is the kernel of a linear map $V' \to \mathbb{F}^{\cdots}$ whose matrix has rows ρ_i. Then also $W := W' \cap V$ is a linear subspace and can again be defined by linear relations. The following function (found in `linear_relations.mag`) computes this subspace in terms of relations.

Function 4 (`TransformRelations`)	
Input:	`E : Fld, rels : SeqEnum[ModTupFldElt]`
Parameters:	-
Output:	`SeqEnum`

Description of function: Transform the sequence `rels` of relations (given as vectors) over a finitely generated field extension of `E` to a sequence of relations over `E`. ◇

This function is used to compute the adjoint spaces in function `HomAdjoints`. The latter function takes a formal desingularization of a surface computed by `ResolveProjectiveSurface` which involves finitely generated field extensions. So the "adjoint constraints" are *a priori* defined w.r.t. these field extensions and have to be translated back to ground field.

```
> Attach("linear_relations.mag");
> Q := Rationals();
> Qi<i> := NumberField(R.1^2+1) where R is PolynomialRing(Q);
> Qis<s> := FunctionField(Qi);
> f := RSpace(Qis,3) ! [(s+1),(i+1),1];
> TransformRelations(Q, [f]);
[
    (1 1 1),
    (0 1 0),
    (1 0 0)
]
```

Assume we have the relation $(s+1)c_1 + (i+1)c_2 + c_3 = 0$ over $\mathbb{Q}[i](s)$. This clearly defines a two-dimensional linear subspace $W' \subset \mathbb{Q}[i](s)^3$. Now assume the c_i take values in $\mathbb{Q}[i]$ then we have $sc_1 + (c_1 + (i+1)c_2 + c_3) = 0$ which is equivalent to $sc_1 = 0$ and $c_1 + (i+1)c_2 + c_3 = 0$, hence, $c_1 = 0$ and $(i+1)c_2 + c_3 = 0$. If the c_i take values in $\mathbb{Q}$ then the last constraint becomes $ic_2 = 0$ and $c_2 + c_3 = 0$, hence $c_2 = c_3 = 0$. Therefore $W := W' \cap \mathbb{Q}^3 = (0,0,0)$. This is also clear from the above output, because the matrix of relations has trivial kernel.

2 Algebraic Power Series

Now we come to the functionality for algebraic power series with fractionary exponents. It can be accessed by attaching the specification `power_series.spec`. Some debugging information can be obtained using the verbose flag `AlgSeries`.

2.1 Data Structures

We represent power series as tuples `<tag, rec>` where `tag` is either the string `"serA"` or `"serB"`. Depending on the value of `tag` the second component `rec` is a record of one of the following two formats:

```
PStypeA := recformat<
defpol : RngUPolElt, // defining polynomial
init   : RngMPolElt, // initial segment of power series root

subs   : SeqEnum,    // opt: - substitutions for variables
mults  : SeqEnum,    //      - multiplicities to adapt lattices
magic  : RngIntElt,  // x-coordinate-order of the linear vertex
Gamma  : Lat,        // (integral) exponent lattice
e      : RngIntElt   // exponent denominator
>;
```

Here `defpol` is a squarefree polynomial vanishing on the represented series and `init` is an initial expansion of the series. In particular, the parent of `defpol`

is a univariate polynomial ring over the parent of `init`. The initial expansion has to be long enough; the meaning of this is explained in [2, Cond. 4.3]. If `subs` is defined, then the actual defining polynomial is `defpol` with variables substituted by the series in `subs`.

The values `mults` and `magic` are quantities that are difficult to explain, but often used by the algorithms and therefore precomputed by the constructors.

Finally `1/e Gamma` is the exponent lattice of the series. It always has to contain $\mathbb{Z}^n$ (*i.e.*, `Gamma` contains $\mathtt{e}\mathbb{Z}^n$). Expansions of the series always have their exponents in `Gamma` and are to be interpreted as being divided by `e` (`MAGMA` doesn't support fractionary exponents).

Remark 2.1 *There are some* ***dangerous pitfalls:***

- *The initial expansion* `init` *has exponents in* `Gamma` *which have to be considered fractionary, i.e., divided by* `e`*. The defining polynomial, instead, has integral exponents (when* `subs` *is empty) or exponents in a lattice dictated by the substituted series in* `subs`*.*
- *`MAGMA` has no mechanism for requiring or checking the type of ordering on a polynomial ring (because the internal methods can change orderings transparently to the user when necessary). The power series package only works, when a* total degree compatible *ordering is chosen, in other words, smaller total degree implies smaller in the ordering. Choose for example* `glex`*-ordering for the parent of* `init`*.*

```
PStypeB := recformat<
series : Tup,        // power series to be substituted
duals  : SeqEnum,    // sequence of vectors in dual lattice

subs   : SeqEnum,    // substitutions for variables
mults  : SeqEnum,    // multiplicities to adapt lattices
magic  : RngIntElt,  // a sort of contraction ratio
Gamma  : Lat,        // (integral) exponent lattice
e      : RngIntElt   // exponent denominator
>;
```

The series represented by this record is obtained as follows: Take the series $\alpha \in \mathbb{E}[\![\underline{x}^{\Gamma_0}]\!]$ represented by `series`. Then compute the image of α under the homomorphism $\underline{x}^{\mathtt{m}} \mapsto \prod_i \mathtt{subs[i]}^{\mathtt{duals[i](m)}}$ where the `subs[i]` are power series with a common domain (but maybe different exponent lattices) and the `duals[i]` are vectors in the dual of Γ_0 (which are integral because Γ_0 must always contain $\mathbb{Z}^n$). There is a further condition on `subs` and `duals` for this homomorphism to be well-defined, see [2, Cond. 4.6].

The values `mults` and `magic` are again precomputed by the constructors. The exponent lattice of the represented series (given by `Gamma` and `e`) is now a common overlattice of the exponent lattices of `subs`.

2.2 Constructors

Using constructors one can construct power series starting from polynomial data or using other power series recursively.

Function 5 (`PolyToSeries`)

Input:	`s : RngMPolElt`
Parameters:	-
Output:	`Tup`

Description of function: Given a multivariate polynomial `s`. Return the series representation of `s`. ◇

Function 6 (`SeriesByRawData`)

Input:	`defpol : RngUPolElt, init : RngMPolElt, Gamma : Lat, e : RngIntElt`
Parameters:	`subs := []`
Output:	`Tup`

Description of function: Define a power series root of a polynomial p using an initial expansion `init` and its exponent lattice $1/$`e Gamma`. The defining polynomial is either `defpol` when `subs` is empty or obtained by substituting the elements of `subs` into the variables of `defpol`. The initial expansion has to be sufficiently long in order to uniquely identify a root, see [2, Cond. 4.3]. ◇

Function 7 (`EvalSeries`)

Input:	`tagged_series : Tup, nu : SeqEnum, v : SeqEnum`
Parameters:	-
Output:	`Tup`

Description of function: Given a series `tagged_series`, a sequence `nu` of vectors in the dual of its exponent lattice and a sequence `v` (of the same length) of power series in some other common domain (with compatible coefficient field). Return the series obtained by substituting $\underline{x}^{\mu} \mapsto \prod_i$ `v[i]`$^{\langle \texttt{nu[i]}, \mu \rangle}$. This requires that `nu` and `v` fulfill a certain condition on the orders, see [2, Cond. 4.6]. ◇

Function 8 (`ImplicitFunction`)

Input:	`defpol : RngUPolElt`
Parameters:	`subs := []`
Output:	`Tup`

Description of function: The unique series with zero constant term defined by a polynomial $p \in \mathbb{E}[x_1, \ldots, x_n][z]$ fulfilling the conditions of the implicit function theorem, *i.e.*, $p(0, \ldots, 0) = 0$ and $\partial p / \partial z(0, \ldots, 0) \neq 0$. The polynomial p is equal to `defpol` possibly substituted with the series in `subs` as in SeriesByRawData. ◇

Function 9 (RationalPuiseux)

Input:	`poly : RngUPolElt`
Parameters:	`Gamma := StandardLattice(0), subs := [], Duval := false, OnlySingular := false, ExtName := "gamma", ExtCount := 0`
Output:	`Tup, SeqEnum, RngIntElt`

Description of function: Given a squarefree, univariate polynomial p over a multivariate polynomial ring, assuming that it is quasi-ordinary, return a complete set of parametrizations (which are rational if `Duval = true`). As in functions ImplicitFunction and SeriesByRawData, p is either equal to `poly` or you can substitute `poly` by specifying a sequence `subs` of series in some common domain (with compatible coefficient field).

The parameter `Gamma` can be used to specify an integral exponent lattice for `poly` which is then considered an element of a polynomial ring with restricted support. Note that this changes the meaning of what a complete set of rational parametrizations means and is allowed only if no sequence `subs` is specified. If `Duval = false` then a complete set of Puiseux series roots of `poly` is returned (up to conjugacy over the coefficient field), *i.e.*, Duval's trick is not applied. If `OnlySingular = true` then some parametrizations corresponding to non-singular branches are not returned.

The first return value is a tuple (Γ, e) and gives the exponent lattice $1/e\Gamma$ of the input polynomial. It is determined (in that order) by the lattices of the series in `subs` (if specified), or the value of `Gamma` (if specified), or it is a standard lattice of the appropriate dimension. The second return value is the sequence of parametrizations.

Parametrizations are represented as tuples (λ, α, n, f) where α is a power series, λ is a sequence of non zero field elements and n and f are positive natural numbers. If $\{\gamma_i\}_i$ is the basis (determined from the representation chosen by `MAGMA`) of the exponent lattice $1/e\Gamma$ then using the homomorphism $\sigma : \underline{x}^{\gamma_i} \mapsto \lambda_i \underline{x}^{\gamma_i}$ we have $\sigma^{\uparrow}(p)(\alpha) = 0$. If `Duval = false` then λ will always be a vector of ones.

The numbers n give the degrees of the lattice extensions and f the degrees of the field extensions used for constructing the parametrizations. These extensions are always minimal but depend on whether Puiseux parametrizations or roots are computed. More precisely, if `Duval = false` then summing over all f-values of the computed roots is equal to the degree of the defining polynomial `poly`. On the other hand, if `Duval = true` then summing over products nf gives the degree.

If the ground field has to be extended, the algebraic elements will be displayed as `ExtName_i` where `i` starts from `ExtCount`. The last return value is the value of `ExtCount` plus the number of field extensions that have been introduced. ◇

We illustrate the constructors by examples. For displaying results we already use the command Expand that will be explained later.

```
> AttachSpec("power_series.spec");
> Q := Rationals(); Qs<s> := FunctionField(Q);
> Qxy<x,y> := PolynomialRing(Q, 2, "glex");
> Qxyz<z> := PolynomialRing(Qxy);
> Qst<t> := PolynomialRing(Qs, 1, "glex");
> Qstu<u> := PolynomialRing(Qst);
```

One can consider polynomials as series.

```
> s0 := PolyToSeries(1 - 3*x + x^2*y + y^20);
> Expand(s0, 10);
true x^2*y - 3*x + 1
```

One can define series by the implicit function theorem at the origin.

```
> s1 := ImplicitFunction(z*(1 - x - y) - x - y);
> Expand(s1, 4);
true x^3 + 3*x^2*y + 3*x*y^2 + y^3 + x^2 + 2*x*y + y^2 + x + y
```

One can define a power series if an initial expansion is known. Note that the following power series has exponent lattice $\mathbb{Z}(\frac{1}{5}, -\frac{2}{5}) + \mathbb{Z}(\frac{2}{5}, \frac{1}{5})$ but its "expansions" are polynomials supported on $\mathbb{Z}(1, -2) + \mathbb{Z}(2, 1)$.

```
> defpol := (1+5*y+10*y^3+10*y^2+5*y^4+y^5)*z^5+(-20*y^3*x-
  30*y^2*x-5*y^4*x-5*x-20*y*x)*z^4+(10*x^2+30*y^2*x^2+10*y^3*x^2+
  30*x^2*y)*z^3+(-20*y*x^3-10*x^3-10*y^2*x^3)*z^2+
  (5*y*x^4+5*x^4)*z-x^5-x^2*y;
> Gamma := Lattice(RMatrixSpace(Integers(), 2, 2) ! [1,-2, 2,1]);
> init := x^2*y;
> s2 := SeriesByRawData(defpol, init, Gamma, 5);
> Expand(s2, 20);
true
-x^2*y^16 + x^5*y^10 + x^2*y^11 - x^5*y^5 - x^2*y^6 + x^5 + x^2*y
```

We can "substitute" series into each other.

```
> X := SeriesByRawData(u^3-t+s*t^2, t, StandardLattice(1), 3);
> Y := PolyToSeries(t);
> duals := [RSpace(Integers(), 2) | [1, 3], [2, 1]];
> s3 := EvalSeries(s2, duals, [X, Y]);
> Expand(s3, 13);
true (-1/9*s^2 - 1/3*s - 1)*t^10 + (-1/3*s + 1)*t^7 + t^4
```

We can compute all the Puiseux series roots of a quasi-ordinary polynomial up to conjugacy over $\mathbb{Q}$.

```
> qopol := z^6 + 3*x*y^2*z^4 + x*y*z^3 + 3*x^2*y^4*z^2 + x^3*y^6;
> _, prms := RationalPuiseux(qopol : Duval := false); prms;
[*
    <[ 1, 1 ], <serA, rec<...>>, 3, 1>,
    <[ 1, 1 ], <serA, rec<...>>, 3, 2>,
    <[ 1, 1 ], <serA, rec<...>>, 3, 1>,
    <[ 1, 1 ], <serA, rec<...>>, 3, 2>
*]
> Domain(prms[2][2]); ExponentLattice(prms[2][2]);
Polynomial ring of rank 2 over Number Field with defining
polynomial $.1^2 - $.1 + 1 over the Rational Field
Graded Lexicographical Order
Variables: x, y
<
    Lattice of rank 2 and degree 2
    Basis:
    ( 1  1)
    ( 1 -2),

    3
>
> Expand(prms[2][2], 15);
true x^3*y^9 + (gamma_0 - 1)*x^2*y^5 + gamma_0*x*y
```

We find that the sum over all field extensions $1+2+1+2 = 6$ is equal to the degree of the defining polynomial `qopol`. The third parametrization involves a field extension of $\mathbb{Q}$ by `gamma_0` s.t. $\texttt{gamma\_0}^2 - \texttt{gamma\_0} + 1 = 0$ and an extension of the exponent lattice to $\mathbb{Z}(\frac{1}{3}, \frac{1}{3}) + \mathbb{Z}(\frac{1}{3}, -\frac{2}{3})$. It turns out that the field extension is not necessary if we are only interested in parametrizations.

```
> _, prms := RationalPuiseux(qopol : Duval := true); prms;
[*
    <[ -1, -1 ], <serA, rec<...>>, 3, 1>,
    <[ -1, 1 ], <serA, rec<...>>, 3, 1>
*]
```

No field extensions have been introduced, but this required the application of automorphisms $\mathbb{Q}[\![x, y]\!] \to \mathbb{Q}[\![x, y]\!]$ in advance (more precisely $x \mapsto -x, y \mapsto -y$ resp. $x \mapsto -x, y \mapsto y$). This time we can sum up the overall extension degrees (*i.e.*, for fields and lattices) $3 \cdot 1 + 3 \cdot 1 = 6$ to the degree of `qopol`.

2.3 Extractors

The following functions provide an interface to conveniently extract information from a power series defined as above.

Function 10 (`Domain`)

Input:	`tagged_series : Tup`
Parameters:	-
Output:	`RngMPol`

Description of function: Return the multivariate polynomial ring that is used for approximating the series by its truncations. ◇

Function 11 (`ExponentLattice`)

Input:	`tagged_series : Tup`
Parameters:	-
Output:	`Tup`

Description of function: Return the exponent lattice $1/e\Gamma$ of the series as tuple (Γ, e) where Γ is an integral lattice and e is an integer. ◇

Function 12 (`Expand`)

Input:	`tagged_series : Tup, ord : RngIntElt`
Parameters:	-
Output:	`BoolElt, RngMPolElt`

Description of function: Given the power series β which is represented by `tagged_series`, let α be the result of substituting variables $x_i \mapsto x_i^e$ where e is taken from the output of `ExponentLattice(tagged_series)` (*i.e.*, α is β without exponent denominators). Return `true` and the truncation of α modulo terms of order greater or equal `ord`. A value `false` indicates that the representation is inconsistent (which should only happen when `RationalPuiseux` is called with non quasi-ordinary input or `SeriesByRawData` is used inconsistently). ◇

Function 13 (`DefiningPolynomial`)

Input:	`tagged_series : Tup`
Parameters:	-
Output:	`RngUPolElt`

Description of function: Return a defining polynomial of the series which is a squarefree univariate polynomial over the multivariate polynomial domain `Domain(tagged_series)`. This computation is expensive and can involve recursive resultant computations. ◇

Function 14 (`Order`)

Input:	`tagged_series : Tup`
Parameters:	`TestZero := false`
Output:	`RngIntElt`

Description of function: Given a series `tagged_series`, return the integral

order of its expansion as returned by Expand, *i.e.*, its fractionary order times the exponent denominator. If tagged_series is zero, this function will *not terminate*. Set TestZero:=true to get a return value -1 in this case, but note that this involves the computationally complex call IsZero(tagged_series). ◇

We can study the series s3.

```
> Domain(s3);
Polynomial ring of rank 1 over Univariate rational function
field over Rational Field
Graded Lexicographical Order
Variables: t
> ExponentLattice(s3);
<
    Standard Lattice of rank 1 and degree 1,

    3
>
> DefiningPolynomial(s3);
(s^45*t^45 - ... - 15*s^30*t^2 - s^30)*u^15 + ... +
(5*s^37*t^41 - ... + 120*s^31*t^19 - 30*s^30*t^18)*u^3 -
s^35*t^40 + ... - 5*s^31*t^21 + s^30*t^20
> Order(s3);
4
```

These commands reveal the following about s3: It is a power series in $\mathbb{Q}(s)[\![t^{1/3}]\!]$, because it is approximated in $\mathbb{Q}(s)[t]$ and has exponent lattice $\frac{1}{3}\mathbb{Z}$. A defining polynomial in $\mathbb{Q}(s)[t][u]$ was also computed. (Recall that s3 has been defined recursively.) The order is $\frac{4}{3}$ which we know already from a previous expansion.

2.4 Arithmetics

Of course we can also do arithmetics on power series.

Function 15 (AlgComb)

Input:	comb : RngMPolElt, series : SeqEnum
Parameters:	-
Output:	Tup

Description of function: Given a polynomial comb in r variables and a sequence series of r power series (in a common domain with compatible coefficient field) return the series obtained by substituting the elements of series for the variables of comb, thus, compute arbitrary algebraic combinations. ◇

Function 16 (`Add`)

Input:	`alpha : Tup, beta : Tup`
Parameters:	-
Output:	`Tup`

Description of function: Add two power series. ◇

Function 17 (`Mult`)

Input:	`alpha : Tup, beta : Tup`
Parameters:	-
Output:	`Tup`

Description of function: Multiply two power series. ◇

One can easily substitute power series into polynomials.

```
> h0 := AlgComb(x^2 + y^2, [s0,s1]);
> Expand(h0, 3);
true 10*x^2 + 2*x*y + y^2 - 6*x + 1
```

This includes of course the ring operations.

```
> h1 := Add(s1, PolyToSeries(One(Qxy)));
> Expand(h1, 4);
true
x^3 + 3*x^2*y + 3*x*y^2 + y^3 + x^2 + 2*x*y + y^2 + x + y + 1
> h2 := Mult(h1, PolyToSeries(1 - x - y));
> Expand(h2, 4);
true 1
> h3 := Add(h2, PolyToSeries(-One(Qxy)));
> Expand(h3, 4);
true 0
```

2.5 Predicates

The following functions provide decision algorithms for algebraic power series. They may involve *recursive resultant computations*, hence, have a high complexity and should be used with care.

Function 18 (`IsZero`)

Input:	`tagged_series : Tup`
Parameters:	-
Output:	`BoolElt`

Description of function: Decides if the series is zero. ◇

Function 19 (`IsEqual`)

Input:	`alpha : Tup, beta : Tup`
Parameters:	-
Output:	`BoolElt`

Description of function: Decides if two series are equal. ◇

Function 20 (`IsPolynomial`)

Input:	`tagged_series : Tup`
Parameters:	-
Output:	`BoolElt, RngMPolElt`

Description of function: Decides whether the series is actually a polynomial (with integral exponents) in the multivariate polynomial domain as returned by `Domain(tagged_series)`. In the positive case also returns that polynomial. This function relies on `SimplifyRep`. ◇

The previous computations suggest that `h2` is 1, in particular it is polynomial (in contrast to `h1`). In this case `h3` would be zero.

```
> IsPolynomial(h1);
false
> IsPolynomial(h2);
true 1
> IsEqual(h2, PolyToSeries(One(Qxy)));
true
> IsZero(h3);
true
```

2.6 Modifiers

The following functions modify the representation of the power series or apply a simple automorphism.

Function 21 (`ScaleGenerators`)

Input:	`tagged_series : Tup, lambda : SeqEnum`
Parameters:	-
Output:	`Tup`

Description of function: Let $\{\gamma_i\}_i$ be the basis (determined from the representation chosen by `MAGMA`) of the exponent lattice of the series `tagged_series`, and let $\sigma : \underline{x}^{\gamma_i} \mapsto$ `lambda[i]`$\underline{x}^{\gamma_i}$. Return the series σ(`tagged_series`). ◇

Function 22 (ChangeRing)

Input:	tagged_series : Tup, S : RngMPol
Parameters:	-
Output:	Tup

Description of function: If S is a multivariate polynomial domain compatible with the approximation domain Domain(tagged_series) return the same power series with new approximation domain S. This is sort of a "coercion between power series rings". ◇

Function 23 (SimplifyRep)

Input:	tagged_series : Tup
Parameters:	Factorizing := true
Output:	Tup

Description of function: "Simplifies" the internal representation of a series. The result will be stored as a record of the format PStypeA without recursive dependencies on other power series. The defining polynomial defpol within this record will be irreducible and therefore a minimal polynomial over Domain(tagged_series). (If Factorizing = false it will be only squarefree.) The next call DefiningPolynomial(tagged_series) returns this polynomial which can be useful (*e.g.*, for IsPolynomial). However, experience shows that the resulting representation is in general neither simple nor more efficient for subsequent computations. ◇

Remark 2.2 *There is a* ***dangerous pitfall:***

Assume we have a series represented by a tree with nodes of type A and B. Assume further that the leaves have been constructed by ***RationalPuiseux*** *with parameter* Gamma *set to some value. Then the intention was probably to work over the subring of a polynomial ring with restricted support. If now* ***SimplifyRep*** *with parameter* Factorizing = true *is called then a minimal polynomial over the whole polynomial ground ring is computed which is maybe not what one wants.*

We can modify s2 by mapping generators (of the Laurent polynomials) $x^{1/5}y^{-2/5} \mapsto 3x^{1/5}y^{-2/5}$ and $x^{2/5}y^{1/5} \mapsto 4x^{2/5}y^{1/5}$.

```
> Expand(ScaleGenerators(s2, [3,4]), 15);
true
64/81*x^2*y^11 - 64/3*x^5*y^5 - 16/9*x^2*y^6 + 48*x^5 + 4*x^2*y
```

One can naturally view h1 as a series in $\mathbb{Q}(i)[\![u, v]\!]$.

```
> Qi<i> := NumberField(R.1^2 + 1) where R is PolynomialRing(Q);
> Qiuv<u,v> := PolynomialRing(Qi, 2, "glex");
```

```
> h4 := ChangeRing(s1, Qiuv);
> Expand(h4, 4); Domain(h4);
true u^3 + 3*u^2*v + 3*u*v^2 + v^3 + u^2 + 2*u*v + v^2 + u + v
Polynomial ring of rank 2 over Qi
Graded Lexicographical Order
Variables: u, v
```

We have seen that the power series `h3` is zero, but its representation does not show this immediately. We can "explicitize" its representation.

```
> SimplifyRep(h3 : Factorizing := true);
<serA, rec<recformat<defpol: RngUPolElt, init: RngMPolElt,
 subs: SeqEnum, mults: SeqEnum, magic: RngIntElt,
 Gamma: Lat, e: RngIntElt> |
    defpol := z,
    init := 0,
    magic := 0,
    Gamma := Standard Lattice of rank 2 and degree 2,
    e := 1
>>
```

3 Formal Desingularization of Surfaces in $\mathbb{P}^3$

This section describes the package developed for formally desingularizing surfaces as introduced in [2]. It can be accessed by attaching the specification `resolve.spec`. Some debugging information can be obtained using the verbose flag `Resolve`.

3.1 Embedded Formal Desingularization of Curves

Before we can specify the following algorithms we have to clarify terminology. Let $C \subset P$ be a plane algebraic curve (where $P = \mathbb{A}^2_{\mathbb{E}}$ or $P = \mathbb{P}^2_{\mathbb{E}}$ for some field $\mathbb{E}$ of characteristic zero) and $\pi : Q \to P$ an *embedded desingularization*, *i.e.*, π is proper birational, Q is regular and $D := \pi^{-1}(C) \subset Q$ is a normal crossing divisor. Further let $\{p_1, \ldots, p_r\} \in Q$ be the generic points of the decomposition of D into irreducible components and $\{q_1, \ldots, q_s\} \in Q$ the closed points of the normal crossings of D.

From π we can construct morphisms $\operatorname{Spec} \widehat{\mathcal{O}_{Q,p_i}} \to P$ and $\operatorname{Spec} \widehat{\mathcal{O}_{Q,q_i}} \to P$. The set of all these morphisms *up to isomorphism of the domain* is called a *formal embedded desingularization* of $C \subset P$. Each of these morphisms has a center on P which is defined to be the image of the closed point.

The two classes of morphisms are represented resp. by homomorphisms $A \to \widehat{\mathcal{O}_{Q,p_i}}$ and $A \to \widehat{\mathcal{O}_{Q,q_i}}$ (where A is either the normal polynomial ring $\mathbb{E}[x, y]$ or the graded polynomial ring $\mathbb{E}[x, y, w]$), and we are free to choose an *isomorphic representation of the codomain*. In other words we represent them by homomorphisms μ_i and ν_i that yield commuting diagrams as follows:

Function 24 (`ResolveCurve`)

Input:	`p : RngMPolElt`
Parameters:	`Factors := [], Ps := 0, Focus := 0, ExtName := "alpha", ExtCount := 0`
Output:	`List, List, List, RngIntElt`

Description of function: Computes essentially a formal embedded resolution of the curve defined by $\mathtt{p} \in \mathbb{E}[x,y]$ (a non-zero bivariate polynomial over a number field) using a succession of point blow ups. Only morphisms whose centers vanish on the ideal generated by `Focus` are considered.

The three returned lists contain elements of the form $(b_1,(y,m_{11}),(p_1,m_{12}))$, $(b_2,(y,m_2))$ and $(b_2,(p_3,m_3))$ respectively. Here b_1, b_2 and b_3 are homomorphisms $\mathbb{E}[x,y] \to \mathbb{E}'[x,y]$ to some bivariate polynomial ring over an algebraic field extension $\mathbb{E}'$ over $\mathbb{E}$.

The first list gathers normal crossings. Precisely, the extended homomorphism $b_1 : \mathbb{E}[x,y] \to \mathbb{E}'[\![x,y]\!]$ corresponds to a ν_i from above. Moreover we have $\langle b_1(\mathtt{p})\rangle = \langle y^{m_{11}} p_1^{m_{12}}\rangle$ where $y=0$ and $p_1=0$ have a normal crossing.

The second list gathers exceptional divisors. The extended homomorphism $b_2 : \mathbb{E}[x,y] \to \mathbb{E}'(x)[\![y]\!]$ corresponds to a μ_i from above. Moreover we have $\langle b_2(\mathtt{p})\rangle = \langle y^{m_2}\rangle$ and $y=0$ corresponds to an exceptional divisor.

Finally, the last list corresponds to the components of the original curve. The extended homomorphism $b_3 : \mathbb{E}[x,y] \to \mathbb{E}'\widehat{[x,y]_{\langle p_3\rangle}}$ (where $\mathbb{E}'=\mathbb{E}$ in this case) corresponds to another μ_i from above. Moreover we have that $b_3(\mathtt{p})$ has multiplicity m_3 in $\mathbb{E}'\widehat{[x,y]_{\langle p_3\rangle}}$ and corresponds to an original curve component.

If known, a factorization of `p` (as returned by the `Factorization`-command of `MAGMA`) can be passed using the parameter `Factors` and the squarefree part of `p` (as returned by the `SquarefreePart`-command) using `Ps`.

If the ground field has to be extended, the algebraic elements will be displayed as `ExtName_i` where `i` starts from `ExtCount`. The last return value is the value of `ExtCount` plus the number of field extensions that have been introduced. ◇

We compute an embedded resolution of an affine plane curve.

```
> AttachSpec("resolve.spec");
> Q := Rationals();
> Qxy<x,y> := PolynomialRing(Q, 2, "glex");
> f := (y^2-x^3)*(x^2-y^2-y^3);
> NCs, EXs, DCs := ResolveCurve(f : Factors := Factorization(f));
> #NCs, #EXs, #DCs;
7 4 2
> NCs[2]; EXs[3]; DCs[1];
[*
```

```
    Mapping from: RngMPol: Qxy to RngMPol: Qxy,
    <y, 4>,
    <-x^2 + 2*x + y, 1>
*]
[*
    Mapping from: RngMPol: Qxy to RngMPol: Qxy,
    <y, 10>
*]
[*
    Mapping from: RngMPol: Qxy to RngMPol: Qxy,
    <y^3 - x^2 + y^2, 1>
*]
> NCs[2][1](x), NCs[2][1](y);
x*y - y
y
```

Here we have passed the factorization of `f` only for illustrative purposes. The curve is the union of a cusp and a node at the origin. It has two singular points over $\mathbb{Q}$, the origin and another intersection point of the two curves which has a residue field of degree 5 over $\mathbb{Q}$.

We have computed the local information of a (not necessarily minimal) embedded resolution and find that it contains of course the 2 components of the strict transform, further 4 exceptional divisors and 7 normal crossings. For example, the pushforward of `f` under the chart map $x \mapsto xy - y, y \mapsto y$ is equal to $y^4(-x^2 + 2x + y)$ up to a local unit. The corresponding germ is isomorphic to a normal crossing in the embedded desingularization. We also see that one of the exceptional divisors has multiplicity 10.

If we were only interested in a local resolution, we would do the following:

```
> NCs, EXs, DCs := ResolveCurve(f : Focus := [x,y]);
> #NCs, #EXs, #DCs;
5 3 0
```

We focus on the origin, hence, do not consider any curve components. We have 1 less exceptional divisor and 2 less normal crossings. This is because the second intersection point of the above two curve components was already a normal crossing, but our algorithm has nevertheless blown it up in the previous example.

Function 25 (`ResolveProjectiveCurve`)

Input:	`p : RngMPolElt`
Parameters:	`Focus := 0, ExtName := "alpha", ExtCount := 0`
Output:	`List, List, List, RngIntElt`

Description of function: Same as `ResolveCurve`, but now `p` is a homogeneous polynomial in three variables that defines a projective curve. Accordingly, the b_j map from the respective homogeneous coordinate ring to some $\mathbb{E}'[x, y]$. ◇

We can also desingularize the projectivization of the above curve.

```
> QXYZ<X,Y,Z> := PolynomialRing(Q, 3);
> F := (Y^2*Z-X^3)*(X^2*Z-Y^2*Z-Y^3);
> NCs, EXs, DCs := ResolveProjectiveCurve(F); #NCs, #EXs, #DCs;
7 4 2
> NCs[3];
[*
    Mapping from: RngMPol: QXYZ to Polynomial ring of rank 2 over
    Rational Field ...,
    <y, 4>,
    <x^2 + 2*x - y, 1>
*]
> NCs[3][1](X);
x*y + y
> NCs[3][1](Y);
y
> NCs[3][1](Z);
1
```

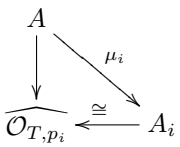
The homomorphisms take a slightly different shape (because they have now $\mathbb{Q}[X, Y, Z]$ as domain), but otherwise they are the same. This is because the curve has no singularities at infinity.

3.2 Formal Desingularization of Surfaces

For curves we have described embedded formal desingularization. For surfaces instead we will need only *formal desingularizations* (*i.e.*, the non-embedded case). Let $S \subset P$ be a hypersurface (where $P = \mathbb{A}^3_{\mathbb{E}}$ or $P = \mathbb{P}^3_{\mathbb{E}}$) and $C \subset S$ a closed subset (which typically contains the singular locus). Further let $\pi : T \to S$ be a *desingularization*, *i.e.*, π is proper birational and T is regular. By $\{p_1, \ldots, p_r\} \in T$ we denote the generic points of the curve components of the decomposition of $D := \pi^{-1}(C)$ into irreducibles.

From π we can construct morphisms $\operatorname{Spec} \widehat{\mathcal{O}_{T,p_i}} \to S$. The set of all these morphisms *up to isomorphism of the domain* is called a *formal desingularization* of S over $C \subset S$. Such a morphism has a center on S which is defined as the image of the closed point (and actually is contained in C).

The morphisms are represented by homomorphisms $A \to \widehat{\mathcal{O}_{T,p_i}}$ (where A is either the algebra $\mathbb{E}[x, y, z]/\langle p \rangle$ or the graded algebra $\mathbb{E}[x, y, z, w]/\langle p \rangle$ with p a defining polynomial), and we are free to choose an *isomorphic representation of the codomain*. In other words we represent them by homomorphisms μ_i that yield commuting diagrams as follows:



The following functions make use of the functions for embedded formal desingularization of curves. But in contrast to them they will return values that involve power series represented as in Section 2.1.

Function 26 (`ResolveAffineMonicSurface`)

Input:	`surf : RngUPolElt`
Parameters:	`Focus := 0, ExtName := "gamma", ExtCount := 0`
Output:	`List, RngIntElt`

Description of function: Given a monic, squarefree polynomial `surf` $\in \mathbb{E}[x,y][z]$ where $\mathbb{E}$ is a number field (*i.e.*, `surf` is univariate over a bivariate polynomial ring). Let $S \subset \mathbb{A}^3_{\mathbb{E}}$ denote the surface defined by it and $C \subset S$ the closed subset defined by $\mathrm{disc}_z(\texttt{surf})$ (*i.e.*, the intersection of S with the cylinder over the discriminant curve when considering the projection $S \to \mathbb{A}^2_{\mathbb{E}}$ in z-direction). Compute a formal desingularization of S over C.

The first return value is a list of elements of the form $((X,Y,Z),o)$ where $X,Y,Z \in \mathbb{F}[\![t]\!]$ are univariate power series (over some field extension $\mathbb{F}$ of transcendence degree 1 over $\mathbb{E}$) s.t. $\texttt{surf}(X,Y,Z) = 0$ and o is an integer. The induced homomorphism $\mathbb{E}[x,y][z]/\texttt{surf} \to \mathbb{F}[\![t]\!]$ corresponds to a μ_i from above and o is its *adjoint order*, *i.e.*, the negation of the order of a special differential form (which is important for computing adjoints, compare Equation (4.1)).

One can specify a focus ideal $\mathcal{F} \subset \mathbb{E}[x,y]$ by passing its generators in `Focus` (compare also the corresponding parameter of `ResolveCurve`). In this case C is taken to be the intersection of S and the cylinder over the zero set of $\mathcal{F} + \langle \mathrm{disc}_z(\texttt{surf}) \rangle$.

If the ground field has to be extended, the algebraic elements will be displayed as `ExtName_i` where `i` starts from `ExtCount`. The last return value is the value of `ExtCount` plus the number of field extensions that have been introduced. A transcendental element will always be displayed as `s`. ◇

We compute a formal desingularization for the affine surface $z^2 - xy = 0$.

```
> AttachSpec("resolve.spec");
> Q := Rationals();
> Qxy<x,y> := PolynomialRing(Q, 2, "glex");
> Qxyz<z> := PolynomialRing(Qxy);
> f := z^2 - x*y;
> desing := ResolveAffineMonicSurface(f); #desing;
3
```

We have computed 3 morphisms. Two of them are centered over the coordinate axes $x = 0$ and $y = 0$. But they might not be of interest, because the surface is normal and has an isolated singularity over the origin.

```
> #ResolveAffineMonicSurface(f : Focus := [x,y]);
1
```

In this example the only remaining morphism corresponds to the exceptional divisor obtained by blowing up the singularity.

Function 27 (ResolveProjectiveSurface)	
Input:	surf : RngMPolElt
Parameters:	AdjComp := false, ExtName := "gamma", ExtCount := 0
Output:	List, RngIntElt

Description of function: Similar to ResolveAffineMonicSurface. Compute a formal desingularization of the projective surface $S \subset \mathbb{P}^3_{\mathbb{E}}$ defined by the squarefree, homogeneous polynomial surf $\in \mathbb{E}[x, y, z, w]$. It will be a formal desingularization over an automatically chosen subset $C \subset S$ (using again the cylinder over the discriminant curve w.r.t. a nice projection onto some $\mathbb{P}^2_{\mathbb{E}}$). Accordingly the elements of the list are now of the form $((X, Y, Z, W), o)$.

If AdjComp = true then only a sublist is returned that is still sufficient for the computation of adjoint spaces (see Section 4.1). The parameters ExtName and ExtCount and the second return value have the usual interpretation. ◇

Computing a formal desingularization is easy.

```
> P<x,y,z,w> := PolynomialRing(Q, 4);
> F := w^3*y^2*z+(x*z+w^2)^3;
> desing := ResolveProjectiveSurface(F); #desing;
26
```

Hence, the formal desingularization of the projective surface defined by F contains 26 morphisms. They are represented by tuples of power series that vanish on F. We have a closer look at the first morphism.

```
> prm, ord := Explode(desing[1]);
> IsZero(AlgComb(F, prm)); ord;
true
4
> X, Y, Z, W := Explode(prm);
> Expand(X, 6); Expand(Y, 6); Expand(Z, 6); Expand(W, 6);
true 1
true -s*t^2
true -t^2
true -1/64*s^2*gamma_0*t^5 + 1/2*gamma_0^2*t^3 + gamma_0*t^2 + t
> Domain(W);
Polynomial ring of rank 1 over Algebraic function field defined
over Univariate rational function field over Rational Field
by $.1^3 - 1/8*s^2
Graded Lexicographical Order
Variables: t
```

One of the morphisms is of type Spec $\mathbb{Q}(s)[\texttt{gamma\_0}][\![t]\!] \to \operatorname{Proj}\mathbb{Q}[x,y,z,w]/\langle \texttt{F}\rangle$ where $\texttt{gamma\_0}^3 - 1/8s^2 = 0$. In particular, $\mathbb{Q}(s)[\texttt{gamma\_0}]$ is isomorphic to the residue field of the corresponding prime divisor on the desingularization. From this one can for example deduce that it is a rational curve. The morphism is given by the ring homomorphism $x \mapsto 1$, $y \mapsto -st^2$, $z \mapsto -t^2$ and $w \mapsto t + \texttt{gamma\_0}t^2 + 1/2\texttt{gamma\_0}^2t^3 - 1/64s^2\texttt{gamma\_0}t^5 + \ldots$.

The adjoint order for this morphism is 4. Consider the chart $x \neq 0$. The special differential form (see Equation (4.1)) in this chart obtained by dehomogenizing is

$$\frac{x^5}{(\partial f/\partial w)(x,y,z,w)}\,\mathrm{d}\,y/x \wedge \mathrm{d}\,z/x.$$

Substituting the values X, Y, Z and W we see that it is mapped to

$$\begin{aligned}
&\frac{\texttt{X}^5}{(\partial f/\partial w)(\texttt{X},\texttt{Y},\texttt{Z},\texttt{W})}\,\mathrm{d}\,\texttt{Y}/\texttt{X} \wedge \mathrm{d}\,\texttt{Z}/\texttt{X}\\
=&\frac{1}{(\partial f/\partial w)(\texttt{X},\texttt{Y},\texttt{Z},\texttt{W})}\,\mathrm{d}(-st^2) \wedge \mathrm{d}(-t^2)\\
=&\frac{1}{(\partial f/\partial w)(\texttt{X},\texttt{Y},\texttt{Z},\texttt{W})}\,(2st\,\mathrm{d}\,t + t^2\,\mathrm{d}\,s) \wedge 2t\,\mathrm{d}\,t\\
=&\frac{1}{(\partial f/\partial w)(\texttt{X},\texttt{Y},\texttt{Z},\texttt{W})}\,2t^3\,\mathrm{d}\,s \wedge \mathrm{d}\,t
\end{aligned}$$

The adjoint order is minus the overall order of this expression, hence, -3 plus the order of $(\partial f/\partial w)(\texttt{X},\texttt{Y},\texttt{Z},\texttt{W})$. We check the computation.

```
> Order(AlgComb(Derivative(F,w), prm));
7
```

If we needed the formal desingularization only in order to compute adjoint spaces we could set the parameter `AdjComp` and forget about some morphisms.

```
> #ResolveProjectiveSurface(F : AdjComp := true);
18
```

4 Surface Parametrization

We have implemented some functionality for the classification and parametrization of algebraic surfaces over characteristic zero. It can be accessed by attaching the specification `classify.spec`. Some debugging information can be obtained using the verbose flag `Classify`.

4.1 Adjoint Computation

Now we describe computation of adjoint spaces. Let $S \subset \mathbb{P}^3_{\mathbb{E}}$ be a surface defined by a homogeneous irreducible polynomial $F \in \mathbb{E}[x_0,x_1,x_2,x_3]$ of degree d and

$\Omega_{\mathbb{E}(S)|\mathbb{E}}$ the vector space of rational differential forms of the function field (over the ground field $\mathbb{E}$ of characteristic zero). We can consider $\Omega_{\mathbb{E}(S)|\mathbb{E}}$ a constant sheaf of $\mathcal{O}_S$-modules. Let $U_i \subset S$ be the affine open subsets of the standard covering w.r.t. this choice of variables.

Let $\omega_S^0 \subset \Omega^{\wedge 2}_{\mathbb{E}(S)|\mathbb{E}}$ be the subsheaf which is locally generated on U_i by

$$\left(\frac{\partial F/\partial x_j}{x_i^{d-1}}\right)^{-1} \bigwedge_{0\le k\le 3 \text{ and } k\notin\{i,j\}} \mathrm{d}\,\frac{x_k}{x_i} \tag{4.1}$$

(for an arbitrary choice of $j \neq i$). By sending this generator to x_i^{d-4} one finds that $\omega_S^0 \cong \mathcal{O}_S(d-4)$. Further let $\mathcal{F}_{S,m} \subset (\Omega^{\wedge 2}_{\mathbb{E}(S)|\mathbb{E}})^{\otimes m}$ be the subsheaf of those forms whose pullbacks are regular on some desingularization of S. It is called the *sheaf of m-adjoints*. It is in fact well-defined, *i.e.*, doesn't depend on any specific desingularization, and one can show $\mathcal{F}_{S,m} \subseteq (\omega_S^0)^{\otimes m}$. For more details we refer to the technical report [3].

Now since $\mathcal{F}_{S,m}$ is a coherent sheaf on the projective scheme $S \subset \mathbb{P}^3_{\mathbb{E}}$ it can be defined by its associated graded module and by the above discussion $\mathcal{F}_{S,m}$ is isomorphic to a subsheaf of $\mathcal{O}_S(m(d-4))$. The graded module corresponding to the latter sheaf is $(\mathbb{E}[x_0,x_1,x_2,x_3]/\langle F\rangle)(m(d-4))$.

Function 28 (`HomAdjoints`)

Input:	`m : RngIntElt, n : RngIntElt, p : RngMPolElt`
Parameters:	`FormalDesing := []`
Output:	`SeqEnum`

Description of function: Given an irreducible homogeneous polynomial $\mathtt{p} \in \mathbb{E}[x_0,\ldots,x_3]$ of degree d (where $\mathbb{E}$ is a number field) and natural numbers `m` and `n` s.t. $\mathtt{n}+\mathtt{m}(d-4) \geq 0$. Let S be the surface in projective space defined by `p`.

Return a basis for the vector space of the degree-`n` homogeneous summand of the graded ring associated to $\mathcal{F}_{S,m}$ (*i.e.*, $\Gamma(S, \mathcal{O}_S(\mathtt{n}) \otimes \mathcal{F}_{S,m})$) as a subspace of the homogeneous forms in $\mathbb{E}[x_0,x_1,x_2,x_3]$ of degree $\mathtt{n}+\mathtt{m}(d-4)$. A precomputed formal desingularization (as returned by a call to `ResolveProjectiveSurface`) may be passed as parameter `FormalDesing`. ◇

We compute adjoint spaces for the surface from above. Therefore we precompute a formal desingularization and use it for several calls to `HomAdjoints`.

```
> AttachSpec("classify.spec");
> Q := Rationals(); P<x,y,z,w> := PolynomialRing(Q, 4);
> F := w^3*y^2*z+(x*z+w^2)^3;
> desing := ResolveProjectiveSurface(F : AdjComp := true);
> HomAdjoints(1, 0, F : FormalDesing := desing);
[]
> HomAdjoints(1, 1, F : FormalDesing := desing);
[
    x*z*w + w^3
]
> HomAdjoints(1, 2, F : FormalDesing := desing);
```

```
[
    x^2*z^2 - w^4, x^2*z*w + x*w^3, x*y*z*w, x*z^2*w + z*w^3,
    x*z*w^2 + w^4, y*z*w^2, y*w^3
]
> HomAdjoints(1, 3, F : FormalDesing := desing);
[
    x^3*z^2 - x*w^4, x^2*y*z^2, x^2*z^3 - z*w^4,
    x^3*z*w + x^2*w^3, x^2*y*z*w, x*y^2*z*w, x^2*z^2*w - w^5,
    x*y*z^2*w, x*z^3*w + z^2*w^3, x^2*z*w^2 + x*w^4, x*y*z*w^2,
    y^2*z*w^2, x*z^2*w^2 + z*w^4, y*z^2*w^2, x*y*w^3, y^2*w^3,
    x*z*w^3 + w^5, y*z*w^3, y*w^4
]
>
> HomAdjoints(2, 0, F : FormalDesing := desing);
[]
> HomAdjoints(2, 1, F : FormalDesing := desing);
[]
> HomAdjoints(2, 2, F : FormalDesing := desing);
[
    x^2*z^2*w^2 + 2*x*z*w^4 + w^6
]
> HomAdjoints(2, 3, F : FormalDesing := desing);
[
    x^3*z^2*w^2 + 2*x^2*z*w^4 + x*w^6, x^2*y*z^2*w^2 - y*w^6,
    x^2*z^3*w^2 + 2*x*z^2*w^4 + z*w^6,
    x^2*z^2*w^3 + 2*x*z*w^5 + w^7, x*y*z^2*w^3 + y*z*w^5,
    x*y*z*w^4 + y*w^6, y^2*z*w^4
]
```

4.2 Classification of Rational Surfaces

We have also implemented the classification of rational surfaces in the sense of Josef Schicho. He enumerates 5 basic cases in [6, p. 17 and Lem. 5.2-5.7] and splits the last case in two, choosing labels "1", "2", "3", "4", "5A" and "5B". The lemmas describing cases 3 and 5A involve a further case distinction. Also a label "0" is useful for the non-rational case. After all, we have the label set

$$\mathcal{L} := \{\text{"0"}, \text{"1"}, \text{"2"}, \text{"3a"}, \text{"3b"}, \text{"4"}, \text{"5Aa"}, \text{"5Ab"}, \text{"5Ac"}, \text{"5B"}\}.$$

Let F be a homogeneous irreducible polynomial in four variables. In each of the above cases the author specifies a set of adjoint spaces defining interesting maps, either birationally to a well-known surface or to a rational normal curve (giving a pencil of rational curves on the surface). More precisely, the maps can be computed from $V_{n,m} :=$ `HomAdjoints`(m, n, F) for certain choices of m and n. Let μ to be the smallest integer s.t. $V_{1,\mu+1} \neq []$. Then the important $V_{n,m}$ for the different cases are as follows:

0	1	2	3a	3b	4
$[]$	$[V_{1,\mu}]$	$[V_{1,\mu}]$	$[V_{2,2\mu+1}, V_{1,\mu}]$	$[V_{2,2\mu+1}]$	$[V_{1,\mu}, V_{2,2\mu-1}]$

5Aa	5Ab	5Ac	5B
$[V_{1,\mu-1}]$	$[V_{1,\mu-1}, V_{2,2\mu-2}]$	$[V_{1,\mu-1}, V_{2,2\mu-2}, V_{3,3\mu-3}]$	$[V_{2,1}]$

The following function does the classification.

Function 29 (`ClassifyProjectiveSurface`)

Input:	`p : RngMPolElt`
Parameters:	`FormalDesing := []`
Output:	`MonStgElt, SeqEnum`

Description of function: Given an irreducible homogeneous polynomial `p` in four variables, defining a projective hypersurface, return the label string in $\mathcal{L}$ identifying the class of the surface and the corresponding sequences $V_{n,m}$ (*cf.* the above table). If a formal desingularization (see `ResolveProjectiveSurface`) is known, it can be passed as parameter `FormalDesing`. ◇

A few examples:

```
> AttachSpec("classify.spec");
> Q := Rationals();
> P<x,y,z,w> := PolynomialRing(Q, 4);
> p1 := x^4 + y^4 - z^2*w^2;
> p2 := 2*x + y + 8*z + 5*w;
> p3 := x^2 - 4*x*z + 3*x*w + y*z - y*w + 2*z^2 - 3*z*w + w^2;
> p4 := (y^2 - w*z)*(w^2 - y*x) + (x*z - y*w)^2;
> p5 := x^3*y - 4*x^3*z - 6*x^3*w - 3*x^2*y^2 - 2*x^2*y*z
        - 3*x^2*y*w + 50*x^2*z^2 + 146*x^2*z*w + 108*x^2*w^2
        - 11*x*y^2*z + 2*x*y^2*w + 61*x*y*z^2 + 149*x*y*z*w
        + 65*x*y*w^2 + 68*x*z^3 + 228*x*z^2*w + 260*x*z*w^2
        + 112*x*w^3 + 4*y^4 - 13*y^3*z - 19*y^3*w + 20*y^2*z^2
        + 77*y^2*z*w + 55*y^2*w^2 + 40*y*z^3 + 106*y*z^2*w
        + 58*y*z*w^2 - 2*y*w^3 + 22*z^4 + 84*z^3*w + 130*z^2*w^2
        + 108*z*w^3 + 38*w^4;
> p6 := x^2*y^2 + 8*x^3*y + 4*x^4 + x*y*z^2 - x^2*z^2 - y^2*w^2
        - 7*x*y*w^2 + 8*x^2*w^2;
> p7 := x^2*w^3 + y^3*w^2 + z^5;
> p8 := w^3*y^2*z + (x*z + w^2)^3;
> ClassifyProjectiveSurface(p1);
0 []
> ClassifyProjectiveSurface(p2);
1 [
    [ y, z, w ]
]
> ClassifyProjectiveSurface(p3);
2 [
    [ x, y, z, w ]
]
> ClassifyProjectiveSurface(p4);
3a [
```

```
    [ x*y - w^2, y^2 - z*w, x*z - y*w ]
    [ x, y, z, w ]
]
> ClassifyProjectiveSurface(p5);
3b [
    [ x^2 - 1114/45*x*z - 1543/45*x*w - 232/15*y*z - 319/15*y*w
      - 241/15*z^2 - 1327/45*z*w - 457/45*w^2,
      x*y - 182/45*x*z - 284/45*x*w - 11/15*y*z - 17/15*y*w
      - 38/15*z^2 - 266/45*z*w - 146/45*w^2,
      -16/45*x*z - 22/45*x*w + y^2 - 28/15*y*z - 61/15*y*w
      - 4/15*z^2 + 32/45*z*w + 92/45*w^2 ]
]
> ClassifyProjectiveSurface(p6);
4 [
    [ x, y ],
    [ x^2, x*y, y^2, x*z, y*z, x*w, y*w ]
]
> ClassifyProjectiveSurface(p7);
5Ac [
    [ z*w^2, w^3 ],
    [ z^2*w^4, y*w^5, z*w^5, w^6 ],
    [ z^3*w^6, y*z*w^7, z^2*w^7, x*w^8, y*w^8, z*w^8, w^9 ]
]
> ClassifyProjectiveSurface(p8);
5B [
    [ x^2*z^2 - w^4, x^2*z*w + x*w^3, x*y*z*w, x*z^2*w + z*w^3,
      x*z*w^2 + w^4, y*z*w^2, y*w^3 ]
]
```

There is also a wrapper that uses `ClassifyProjectiveSurface` for a given input and calls an according subroutine to parametrize the corresponding surface type. Some of these subroutines are presented in the following sections.

Function 30 (`ParametrizeProjectiveHypersurface`)

Input:	`X : Sch, P2 : Sch`
Parameters:	`FormalDesing := []`
Output:	`BoolElt, MapSch`

Description of function: Given an integral hypersurface $\texttt{X} \subset \mathbb{P}^3_{\mathbb{Q}}$ and a projective plane `P2` over $\mathbb{Q}$. Return `false` if the surface is not rational over $\mathbb{Q}$, otherwise return `true` and a parametrization $\texttt{P2} \dashrightarrow \texttt{X}$. This routine might produce an error if the surface type is not yet implemented (compare Section 4.5). If a formal desingularization (see `ResolveProjectiveSurface`) is known, it can be passed as parameter `FormalDesing`. ◇

Function 31 (ParametrizeProjectiveSurface)

Input:	X : Sch, P2 : Sch
Parameters:	
Output:	BoolElt, MapSch

Description of function: Given an integral surface X $\subset \mathbb{P}^n_{\mathbb{Q}}$ for some $n \geq 2$ and a projective plane P2 over $\mathbb{Q}$. Return false if the surface is not rational over $\mathbb{Q}$, otherwise return true and a parametrization P2 --> X. This routine might produce an error, see ParametrizeProjectiveHypersurface. ◇

Function 32 (IsRational)

Input:	X : Sch
Parameters:	FormalDesing := []
Output:	BoolElt

Description of function: Given an integral hypersurface X $\subset \mathbb{P}^3_{\mathbb{E}}$ over some number field. Return whether the surface is rational over the algebraic closure $\overline{\mathbb{E}}$. If a formal desingularization (see ResolveProjectiveSurface) is known, it can be passed as parameter FormalDesing. ◇

Continuation of the examples:

```
> P2<X,Y,W> := ProjectiveSpace(Q, 2);
> P3 := ProjectiveSpace(P);
> ParametrizeProjectiveHypersurface(Scheme(P3, p1), P2);
false
> ParametrizeProjectiveHypersurface(Scheme(P3, p2), P2);
true Mapping from: Prj: P2 to Scheme over Rational Field
defined by 2*x + y + 8*z + 5*w
with equations :
-1/2*X - 4*Y - 5/2*W
X
Y
W
> ParametrizeProjectiveHypersurface(Scheme(P3, p3), P2);
true Mapping from: Prj: P2 to Scheme over Rational Field
defined by x^2 - 4*x*z + 3*x*w + y*z - y*w + 2*z^2 - 3*z*w + w^2
with equations :
X^2 - 2*X*W
X^2 + X*Y - 4*X*W - Y*W + 2*W^2
X^2 - 3*X*W + Y*W
X^2 - 4*X*W + Y*W + 2*W^2
> ParametrizeProjectiveHypersurface(Scheme(P3, p4), P2);
true Mapping from: Prj: P2 to Scheme over Rational Field
defined by x^2*z^2 - x*y^3 - x*y*z*w + 2*y^2*w^2 - z*w^3
with equations :
2*X^2*Y^2*W^2 - Y*W^5
X^4*Y^2 - X^2*Y*W^3
```

```
X^3*Y*W^2
X^3*Y^2*W
> ParametrizeProjectiveHypersurface(Scheme(P3, p5), P2);
true Mapping from: Prj: P2 to Scheme over Rational Field
defined by ...
with equations :
-1/4*X^2 + 9/2*X*Y + 1/2*X*W - 69/4*Y^2 - 87/8*Y*W + 1/2*W^2
1/2*X*Y + 11/4*X*W - 5/8*Y^2 - 131/8*Y*W - 63/8*W^2
-11/8*X*Y + 7/8*X*W + 13/4*Y^2 - 11/2*W^2
X*Y - 1/8*X*W - 23/8*Y^2 + 4*W^2
> ParametrizeProjectiveHypersurface(Scheme(P3, p6), P2);
true Mapping from: Prj: P2 to Scheme over Rational Field
defined by 4*x^4 + 8*x^3*y + x^2*y^2 - x^2*z^2 + 8*x^2*w^2
            + x*y*z^2 - 7*x*y*w^2 - y^2*w^2
with equations :
-3/8*X^2*Y^2*W - 2*X^2*Y*W^2 - 8/3*X^2*W^3 - 59/24*X*Y^2*W^2
    - 38/3*X*Y*W^3 - 16*X*W^4 + 17/6*Y^2*W^3 + 44/3*Y*W^4
    + 56/3*W^5
-3/8*X^3*Y^2 - 2*X^3*Y*W - 8/3*X^3*W^2 - 59/24*X^2*Y^2*W
    - 38/3*X^2*Y*W^2 - 16*X^2*W^3 + 17/6*X*Y^2*W^2
    + 44/3*X*Y*W^3 + 56/3*X*W^4
-1/8*X^2*Y^2*W - 4/3*X^2*Y*W^2 - 8/3*X^2*W^3 - 1/12*X*Y^2*W^2
    - 16/3*X*Y*W^3 - 40/3*X*W^4 + 19/3*Y^2*W^3 + 104/3*Y*W^4
    + 48*W^5
-3/8*X^2*Y^2*W - 2*X^2*Y*W^2 - 8/3*X^2*W^3 - 8/3*X*Y^2*W^2
    - 14*X*Y*W^3 - 56/3*X*W^4 + Y^2*W^3 + 20/3*Y*W^4 + 32/3*W^5
> ParametrizeProjectiveHypersurface(Scheme(P3, p7), P2);
Runtime error in 'ParametrizeDelPezzoDeg5': Parametrization of
degree 5 Del Pezzos is not yet implemented!
> ParametrizeProjectiveHypersurface(Scheme(P3, p8), P2);
Runtime error in 'ParametrizeDelPezzoDeg6': Parametrization of
degree 6 Del Pezzos is not yet implemented!
> IsRational(Scheme(P3, p7)), IsRational(Scheme(P3, p8));
true true
```

The last two examples do not work because Del Pezzos of degrees 5 to 8 can not yet be parametrized. Nevertheless we can determine that the example surfaces are rational, at least over $\overline{\mathbb{Q}}$. Finally two none-hypersurfaces:

```
> P<u,v,w,x,y> := PolynomialRing(Q, 5);
> P2<X,Y,Z> := ProjectiveSpace(Q, 2);
> S := Scheme(ProjectiveSpace(P),
  ideal<P | u^2 + v^2 + w^2 - x^2, y - x>);
> ParametrizeProjectiveSurface(S, P2);
true Mapping from: Prj: P2 to Sch: S
with equations :
2*X*Z
-X^2 - Y^2 + Z^2
```

```
2*Y*Z
X^2 + Y^2 + Z^2
X^2 + Y^2 + Z^2
> ParametrizeProjectiveSurface(P2, P2);
true Mapping from: Prj: P2 to Prj: P2
with equations :
X
Y
Z
```

For convenience we also give a purely algebraic version that avoids schemes.

Function 33 (`Solve`)

Input:	`p : RngMPolElt, F : FldFunRat`
Parameters:	-
Output:	`SeqEnum`

Description of function: Given a polynomial $\mathtt{p} \in \mathbb{Q}[x,y,z]$ a field $\mathtt{F} = \mathbb{Q}(u,v)$. Return a sequence of triples $(X,Y,Z) \in \mathtt{F}^3$ s.t. $\mathtt{p}(X,Y,Z) = 0$ and the Jacobian $\partial(X,Y,Z)/\partial(u,v)$ has generically rank 2. Moreover every other such solution is obtained by applying an endomorphism of `F`. (This routine might produce an error if the corresponding surface type is not yet implemented, see above.) $\diamond$

```
> P<x,y,z> := PolynomialRing(Q, 3);
> F<s,t> := RationalFunctionField(Q, 2);
> p := (x^4+y^4-z^2)*(2*x + y + 8*z + 5)
       *(x^2 - 4*x*z + 3*x + y*z - y + 2*z^2 - 3*z + 1);
> Solve(p, F);
[
    [ -1/2*s - 4*t - 5/2, s, t ],
    [
        (s^2 - 2*s)/(s^2 - 4*s + t + 2),
        (s^2 + s*t - 4*s - t + 2)/(s^2 - 4*s + t + 2),
        (s^2 - 3*s + t)/(s^2 - 4*s + t + 2)
    ]
]
```

4.3 Quadric Surfaces

We have implemented an algorithm for proper rational parametrization of quadric hypersurfaces of $\mathbb{P}^3$ which correspond to class 2 in the above classification.

Function 34 (`ParametrizeQuadric`)

Input:	`X : Sch, P2 : Sch`
Parameters:	-
Output:	`BoolElt, MapSch`

Description of function: Given a quadric projective hypersurface `X` $\subset \mathbb{P}^3_{\mathbb{Q}}$ and a projective plane `P2`. Return `false` if `X` is not parametrizable over the rationals, otherwise return `true` and a parametrization `P2 --> X`. $\diamond$

A few examples:

```
> AttachSpec("classify.spec");
> Q := Rationals();
> P2<X,Y,W> := ProjectiveSpace(Q, 2);
> P3<x,y,z,w> := ProjectiveSpace(Q, 3);
> X1 := Scheme(P3, x^2 + y^2 + z^2 + w^2);
> X2 := Scheme(P3, x^2 + y^2 + z^2 - w^2);
> X3 := Scheme(P3, x^2 + y^2 + z^2);
> X4 := Scheme(P3, x^2 + y^2 - z^2);
> X5 := Scheme(P3, x^2 - 4*x*z + 3*x*w + y*z - y*w + 2*z^2
                   - 3*z*w + w^2);
> ParametrizeQuadric(X1, P2);
false
> ParametrizeQuadric(X2, P2);
true Mapping from: Prj: P2 to Sch: X2
with equations :
2*X*W
2*Y*W
-X^2 - Y^2 + W^2
X^2 + Y^2 + W^2
> ParametrizeQuadric(X3, P2);
false
> ParametrizeQuadric(X4, P2);
true Mapping from: Prj: P2 to Sch: X4
with equations :
X*Y
-1/2*X^2 + 1/2*Y^2
1/2*X^2 + 1/2*Y^2
Y*W
> ParametrizeQuadric(X5, P2);
true Mapping from: Prj: P2 to Sch: X5
with equations :
X^2 - 2*X*W
X^2 + X*Y - 4*X*W - Y*W + 2*W^2
X^2 - 3*X*W + Y*W
X^2 - 4*X*W + Y*W + 2*W^2
```

The surfaces `X1` and `X3` are not rational over $\mathbb{Q}$ but only over $\mathbb{Q}[i]$.

4.4 Ruled Surfaces

We have implemented an algorithm for proper rational parametrization of surfaces with a rational pencil (along the lines of [7]) which correspond to classes 3a and 4 in the above classification.

Function 35 (ParametrizePencil)

Input:	pencil : MapSch, P2 : Sch
Parameters:	-
Output:	BoolElt, MapSch

Description of function: Given a ruled surface X as the domain of a rational pencil `pencil` defined over $\mathbb{Q}$ (*i.e.*, a rational map X `--→` $\mathbb{P}^n_{\mathbb{Q}}$ for some n with image a rational normal curve) and a projective plane `P2`. Return `false` if `X` is not parametrizable over the rationals, otherwise return `true` and a parametrization `P2 --→` X. ◇

An example:

```
> AttachSpec("classify.spec");
> Q := Rationals();
> P3<x,y,z,w> := ProjectiveSpace(Q, 3);
> P2<X,Y,Z> := ProjectiveSpace(Q, 2);
> X := Scheme(P3, x^2*z^2 - x*y^3 - x*y*z*w + 2*y^2*w^2 - z*w^3);
> pencil := map<X -> P2 | [x*y - w^2, y^2 - z*w, x*z - y*w]>;
> DefiningPolynomial(Image(pencil));
X*Y - Z^2
> ParametrizePencil(pencil, P2);
true Mapping from: Prj: P2 to Sch: X
with equations :
-X^4*Y + 2*X^2*Z^3
-X^2*Y*Z^2 + Z^5
X*Y*Z^3
X*Z^4
```

4.5 Del Pezzo Surfaces

The following function is only a partial implementation and the reason why also ParametrizeProjectiveSurface is not complete. It should parametrize Del Pezzo surfaces which correspond to classes 5Aa, 5Ab, 5Ac and 5B in the above classification.

Del Pezzo surfaces are further classified by their degree which lies between 1 and 9, where Del Pezzos of degree 1 and 2 are naturally embedded in some weighted projective space and Del Pezzos of higher degree are embedded in an ordinary projective space of dimension equal to the degree of the surface. The file `delpezzo.mag` contains routines for reducing Del Pezzo surfaces to a degree between 5 and 9. Since this algorithm is not described (in detail) elsewhere we have attached a short description in Appendix A. Del Pezzos of higher degree should be parametrized separately by the methods in `delpezzo_degrees.mag`.

Function 36 (`ParametrizeDelPezzo`)

Input:	`X : Sch, P2 : Sch`
Parameters:	-
Output:	`BoolElt, MapSch`

Description of function: Given a Del Pezzo surface `X` embedded in some (possibly weighted) projective space and a projective plane `P2`. Return `false` if `X` is not parametrizable over the rationals, otherwise return `true` and a parametrization `P2 --> X`. In many cases, though, this function will return an error indicating that its implementation is not complete. ◇

List of Functions

An alphabetical list of all defined intrinsics. Underlined page numbers indicate the page where the identifier is defined, others a page where it is referenced.

References

[1] Tobias Beck, *Efficient Surface Tool Box – A* MAGMA *Software Package –*, http://www.ricam.oeaw.ac.at/software/symcomp/surfaces.tar.gz, November 2007. 1

[2] ———, *Formal Desingularization of Surfaces – The Jung Method Revisited –*, Tech. Report 2007-31, RICAM, December 2007. 6, 7, 16

[3] Tobias Beck and Josef Schicho, *Adjoint Computation for Hypersurfaces Using Formal Desingularizations*, Tech. Report 2008-2, RICAM, January 2008. 23

[4] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265, Computational algebra and number theory (London, 1993). 1

[5] Jana Pílniková, MAGMA *software package for finding Diophantine solutions of quadratic forms*, http://www.risc.uni-linz.ac.at/research/alggeom/software/download/qf.tar.gz, July 2005. 1

[6] Josef Schicho, *Rational parametrization of surfaces*, J. Symbolic Comput. **26** (1998), no. 1, 1–29. 24

[7] ———, *Proper parametrization of surfaces with a rational pencil*, Proceedings of the 2000 International Symposium on Symbolic and Algebraic Computation (St. Andrews) (New York), ACM, 2000, pp. 292–300 (electronic). 30

A Reduction of Del Pezzo Surfaces

Let $\mathbb{E}$ be a field of characteristic zero and $\overline{\mathbb{E}}$ its algebraic closure. The following method is due to Josef Schicho. Note that this appendix is only an early draft.

First we recall some results that will be important in the sequel. We just state the results, but do not give (detailed) proofs or references. Let X be a normal surface over $\mathbb{E}$ and $\pi_X : \widetilde{X} \to X$ its minimal desingularization. Then for each divisor $E \in \mathrm{Div}(X)$ we define its pullback $\widetilde{E} := \pi_X^*(E) \in \mathrm{Div}(\widetilde{X})$ and its strict transform $\overline{E} \in \mathrm{Div}(\widetilde{X})$ (*i.e.* the closure of the pullback of an open subset of E that doesn't meet the singular locus). Intersection products are always computed on the desingularization; If $D \in \mathrm{Div}(X)$ is another divisor we set $DE := \widetilde{D}\widetilde{E}$. The *canonical sheaf* on X is defined as $\omega_X := \pi_*(\omega_{\widetilde{X}})$ where $\omega_{\widetilde{X}}$ is the canonical sheaf on $\widetilde{X}$. Then X is called a *Del Pezzo surface* iff

1. $X \times \mathrm{Spec}\,\overline{\mathbb{E}}$ is *birationally trivial*, *i.e.* birationally equivalent to $\mathbb{P}^2_{\overline{\mathbb{E}}}$,
2. the canonical sheaf ω_X is *invertible* and its *dual* $\omega_X^\vee$ is *ample*.

Then some power of $\omega_X^\vee$ is *very ample* and so $\omega_X^\vee$ defines a natural embedding into a weighted projective space which is called the *anticanonical embedding*. Let $K \in \mathrm{Div}(X)$ be the divisor associated to ω_X. For a Del Pezzo surface one can define its *degree* as the intersection product $\deg(X) := K^2 = (-K)^2$. Then $1 \leq \deg(X) \leq 9$ and $\deg(X)$ coincides with the projective degree under the anticanonical embedding.

Let $E \subset X$ be an irreducible curve and $\overline{E} \subset \widetilde{X}$ its strict transform under π_X. We call E an *exceptional curve* if $\overline{E}$ can be blown down smoothly. Then there is a commutative diagram

$$\begin{array}{ccc} \widetilde{X} & \xrightarrow{\widetilde{\phi}} & \widetilde{X}' \\ {\scriptstyle \pi_X}\downarrow & & \downarrow{\scriptstyle \pi_{X'}} \\ X & \overset{\phi}{\dashrightarrow} & X' \end{array}$$

where $\widetilde{\phi}$ is the blow down of $\overline{E}$ and ϕ is an induced rational map. Assuming that $E \times \mathrm{Spec}\,\overline{\mathbb{E}}$ falls apart into r distinct curves, X' is again a Del Pezzo surface and $\deg(X') = \deg(X) + r$. So we can reduce to higher degrees if exceptional curves exists and as usual we call a Del Pezzo surface *minimal* if it does not contain any exceptional curves.

Further $\widetilde{\phi}^*(\widetilde{K}') = \widetilde{K} - \overline{E}$ and $\pi_{X'}$ is induced by $-\widetilde{K}'$. Then $\phi \circ \pi_X$ is induced by $\widetilde{\phi}^*(-\widetilde{K}') = -\widetilde{K} + \overline{E}$. In order to compute $\widetilde{\phi}$, resp. ϕ, we have to find the linear system of $-\widetilde{K} + \overline{E}$ (and maybe of its multiples), resp. the image on X.

It turns out that if $\mathbb{E} = \overline{\mathbb{E}}$ then X is only minimal if $\deg(X) > 7$. On the other hand if $\mathbb{E} = \mathbb{Q}$ and X is birationally trivial already over $\mathbb{Q}$ then minimality implies only $\deg(X) > 4$. So for parametrizing over $\mathbb{Q}$ one has to provide

1. methods to parametrize Del Pezzos of degrees 5 to 9 (implementations should go to `delpezzo_degrees.mag`) and
2. to reduce Del Pezzos to higher degrees when possible over the ground field (see `delpezzo.mag`).

In this appendix we describe how to treat the second item. More precisely, we will show how to find exceptional curves and how to blow them down.

Geometry

The anticanonical embedding for a Del Pezzo X of degree 1 maps to the weighted projective space $\mathbb{P}_{\mathbb{E},(1,1,2,3)}$, for degree 2 it maps to $\mathbb{P}_{\mathbb{E},(1,1,1,2)}$ and for higher degrees $\omega_X^\vee$ is already very ample and X is embedded in ordinary projective space $\mathbb{P}^d_{\mathbb{E}}$ with $d := \deg(X)$.

Del Pezzos of Degree 1

Let $X \subset \mathbb{P}_{\mathbb{E},(1,1,2,3)}$ be a Del Pezzo of degree 1 (anticanonically embedded). We use coordinates x, y, z, w with corresponding degrees. Then X has a single defining equation $F(x, y, z, w)$ of (weighted) degree 6 which contains w^2, and z^3 appears with a non-zero coefficient. Hence $F = w^2 + F_3(x, y, z)w + F_6(x, y, z)$ where F_3 is of degree 3 and F_6 is of degree 6. Using a Tschirnhausen transformation $w \mapsto w - \frac{1}{2}F_3$ we can get rid of the linear term in w, *i.e.*, $F_3 = 0$. Hence, w.l.o.g. we may assume that $F = w^2 + cz^3 + F_2(x, y)z^2 + F_4(x, y)z + F_6(x, y)$ with analogous conventions for the degrees and $c \neq 0$. We may even assume that $c = 1$, otherwise multiply by c^2 and substitute $w \mapsto \frac{1}{c}w$, $z \mapsto \frac{1}{c}z$. By a further Tschirnhausen transformation $z \mapsto z - \frac{1}{3}F_2(x, y)$ we can get rid of the quadratic term in z. Altogether we may assume that $F = w^2 + z^3 + F_4(x, y)z + F_6(x, y)$ (absolutely irreducible) where F_4 and F_6 are polynomials of (ordinary) degrees 4 and 6 respectively.

When dealing with Del Pezzos of degree 1 we will use two projection maps

$$\begin{aligned}\varphi_1 &: X \to \mathbb{P}_{\mathbb{E},(1,1,2)} : (x : y : z : w) \mapsto (x : y : z) \text{ and}\\ \varphi_2 &: \mathbb{P}_{\mathbb{E},(1,1,2)} \dashrightarrow \mathbb{P}_{\mathbb{E},(1,1)} \cong \mathbb{P}^1_{\mathbb{E}} : (x : y : z) \mapsto (x : y).\end{aligned}$$

Note that $\mathbb{P}_{\mathbb{E},(1,1,2)}$ is isomorphic to a cone $C \subset \mathbb{P}^3_{\mathbb{E}}$ with coordinates r, s, t, u, defining equation $rt - s^2$ and isomorphism given by $(x : y : z) \mapsto (x^2 : xy : y^2 : z)$. Then the vertex has coordinates $(x : y : z) = (0 : 0 : 1)$ and φ_2 is the projection to the base. Hence φ_2 is defined everywhere, but in the vertex. On the other hand φ_1 is a finite morphism and 2 to 1.

Del Pezzos of Degree 2

Let $X \subset \mathbb{P}_{\mathbb{E},(1,1,1,2)}$ be a Del Pezzo of degree 2 with coordinates x, y, z, w. Using a Tschirnhausen transformation as above we may assume that X is given by a single (absolutely irreducible) defining equation $F(x, y, z, w) = w^2 + F_4(x, y, z)$ where F_4 is a polynomial of (ordinary) degree 4.

Here we use only the projection map

$$\varphi : X \to \mathbb{P}_{\mathbb{E},(1,1,1)} \cong \mathbb{P}^2_{\mathbb{E}} : (x : y : z : w) \mapsto (x : y : z)$$

which is a finite morphism and 2 to 1.

Del Pezzos of Degree 3 and Higher

Let $X \subset \mathbb{P}^d_{\mathbb{E}}$ be a Del Pezzo of degree $d > 2$. Using coordinates $x_0, \ldots, x_d$, X is defined by a homogeneous (absolutely) prime ideal $I \subset \mathbb{E}[x_0, \ldots, x_d]$.

Finding Contractible Lines

Assume for the following sections that $\mathbb{E} = \overline{\mathbb{E}}$ is algebraically closed. Let $D \subset X$ be a hyperplane section (then $\widetilde{D}$ is linearly equivalent to $-\widetilde{K}$). An irreducible curve $E \subset X$ is exceptional iff

$$1)\ \overline{E}^2 = -1 \qquad \text{and} \qquad 2a)\ \overline{E}\widetilde{K} = -1.$$

Since $\widetilde{E} = \overline{E} + R$ and R is contracted to an at most finite set of singular points, we could have chosen D (possibly using a suitable reembedding and considering nD instead) such that it doesn't meet these points and we have have $\widetilde{D}R = 0$. Then $DE = \widetilde{D}\widetilde{E} = \widetilde{D}(\overline{E} + R) = \widetilde{D}\overline{E} = -\widetilde{K}\overline{E}$, so the second condition is equivalent to

$$2b)\ DE = 1,$$

i.e., E has to be "a line" in the sense of a degree one curve.

Del Pezzos of Degree 1

The elements of $|D|$ are exactly the fibers of points under $\varphi_2 \circ \varphi_1$. The fibers of φ_2 are lines on the cone C connecting vertex and a point on the base. In other words $D_0 := \varphi_1(D) \subset \mathbb{P}_{\mathbb{E},(1,1,2)}$ is such a line. Now $E_0 := \varphi_1(E) \subset \mathbb{P}_{\mathbb{E},(1,1,2)}$ has to be an irreducible curve with $E_0 D_0 = 1$ (because D is a full hyperplane section and so $D_0 E_0 > 1$ would also imply $DE > 1$). We have one of the following:

1. E_0 is a hyperplane section of the cone C not containing the vertex. Then we must have $\varphi_1^*(E_0) = E_1 + E_2$; Indeed, if $\varphi_1^*(E_0) = E$ then $DE = 2$ because $D_0 E_0 = 1$ and φ_1 is 2 to 1.

   If $\varphi_1^*(E_0)$ decomposes then $DE_i = 1$ for $i \in \{1, 2\}$. Further E_i, and hence $\overline{E_i}$, is a rational curve (a parametrization is given in the description of the implementation below). The genus formula gives

   $$\overline{E_i}^2 - 1 = \overline{E_i}^2 - E_i D = \overline{E_i}^2 + \overline{E_i}\widetilde{K} = 2\,\mathrm{g}(\overline{E_i}) - 2 = 0 - 2 = -2$$

   where the second equality follows from the equivalence of the conditions 2a) and 2b). So $\overline{E_i}^2 = -1$ and these contractible lines appear in complementary pairs.

2. E_0 is itself a line in C through the vertex and a point on the base. Then E_0 would be defined by $ax + by$ with, say, $b \neq 0$. Substituting into F we find that the fiber of E_0 is isomorphic to a curve in $\mathbb{P}_{\mathbb{E},(1,2,3)}$ given by $F|_{y=-(a/b)x} = w^2 + z^3 + \cdots \in \mathbb{E}[x, z, w]$ which is obviously irreducible. Hence in this case we must have $E = \varphi_1^*(E_0)$ is linearly equivalent to D. Therefore we have $DE = D^2 = 1$ (the degree of the Del Pezzo).

   The ramification locus $B \subset \mathbb{P}_{\mathbb{E},(1,1,2)}$ of φ_1 is given by $F|_{w=0}$. If E_0 doesn't pass through a singularity of the curve B then E doesn't pass through singular points of X. So $\widetilde{E} = \overline{E} \cong E$, we have $\overline{E}^2 = \widetilde{E}^2 = E^2 = DE = 1$ and E is not contractible.

   On the other hand assume that E_0 passes through a singular point $p \in B$. Now E passes through the singular point $q := (\varphi_1)^{-1}(p) \in X$ and so $\widetilde{E} = \overline{E} + R$ where R is contracted to q. We know already

   $$1 = DE = -\widetilde{K}\overline{E} = \widetilde{D}\overline{E} = \widetilde{E}\overline{E} = (\overline{E} + R)\overline{E} = \overline{E}^2 + \overline{E}R.$$

We have to show $\overline{E}R = 2$ to get $\overline{E}^2 = -1$. R results from resolving q which has multiplicity greater 1. The curve $\overline{E}$ is the strict transform of E which contains q. Therefore $\overline{E}R > 1$. On the other hand the genus formula gives

$$2\,\mathrm{g}(\overline{E}) - 2 = \overline{E}^2 + \overline{E}\widetilde{K} = \overline{E}^2 - \overline{E}\widetilde{D} = \overline{E}^2 - \overline{E}(\overline{E} + R) = -\overline{E}R$$

and hence also $\overline{E}R \leq 2$. So such contractible E appears isolated.

Contractible Lines in Pairs: We consider the first case for E_0. To find an exceptional curve we have to look for hyperplane sections of the cone C s.t. the preimage in X under φ_1 decomposes into two components E_1 and E_2. Let $z - (ax^2 + bxy + cy^2)$ be the equation of a hyperplane section of C; it is the pullback of $u-(ar+bs+ct)$. We substitute in F to get $F|_{z=ax^2+bxy+cy^2} = w^2-G$ where $G \in \mathbb{E}[a,b,c][x,y]$ is a polynomial of (ordinary) degree 6 in x, y. Then the preimage of the hyperplane section decomposes for special values of a, b, c if and only if $F|_{z=ax^2+bxy+cy^2}$ factors if and only if G is a square. The last condition can be expressed by polynomial conditions on the coefficients a, b, c. More precisely, we can derive an at most zero-dimensional polynomial ideal $I \subset \mathbb{E}[a,b,c]$. Assume that (a_0, b_0, c_0) is a root of I and H a square root of $G|_{a=a_0,b=b_0,c=c_0}$. Then the morphisms $(x : y) \mapsto (x : y : a_0x^2+b_0xy+c_0y^2 : \pm H)$ parametrize two contractible lines on X.

Isolated Contractible Lines: Now consider the second case for E_0. We find all points $(a_0 : b_0 : c_0) \neq (0 : 0 : 1)$ in which $B \subset \mathbb{P}_{\mathbb{E},(1,1,2)}$ (given by $F|_{w=0}$) is singular. Then E_0 has equation $b_0x - a_0y$ (the line through $(a_0 : b_0 : c_0)$ and $(0 : 0 : 1)$) and E is the only curve above E_0. Later we show that for blowing down E we actually need the complement R in the decomposition $\widetilde{E} = \overline{E} + R$ which in this case is contracted to the point $(a_0, b_0, c_0, 0) \in X$. Actually this point is already sufficient for computing the contraction map.

Del Pezzos of Degree 2

The elements of $|D|$ now are the fibers of lines (in $\mathbb{P}^2_{\mathbb{E}}$) under φ. Hence $D_0 := \varphi(D) \subset \mathbb{P}_{\mathbb{E},(1,1,2)}$ is such a line. We look for an exceptional curve E. Let $E_0 := \varphi(E)$. As above we must have $E_0D_0 = 1$ and E_0 is also a line, say, with equation $z - (ax + by)$ (here also $x - \ldots$ and $y - \ldots$ are possible). Let $F|_{z=ax+by} = w^2 - G$ with $G \in \mathbb{E}[x,y]$. We have one of the following:

1. If G is no square then $F|_{z=ax+by} \in \mathbb{E}[x,y,w]$ is irreducible and $E = \varphi^*(E_0) \in |D|$ is a full hyperplane section. Therefore we would have $DE = D^2 = 2$ (the degree of the Del Pezzo) and E can not be contractible.

2. If $G = H^2$ then $F|_{z=ax+by} = (w + H)(w - H)$, $\varphi^*(E_0) = E_1 + E_2$ decomposes and $DE_i = 1$ for $i \in \{1, 2\}$. Further E_i, and hence $\overline{E_i}$, is a rational curve (a parametrization is given in the description of the implementation below). As before this implies $\overline{E_i}^2 = -1$. So such a line is contractible and appears in a complementary pair.

Del Pezzos of Degree 3 and Higher

We look for a curve $E \subset \mathbb{P}^d_{\mathbb{E}}$ on X s.t. $DE = 1$. By repeatedly applying the arguments above (projecting to suitable linear subspaces) we find that E must be the intersection of $d-1$ hyperplanes given by linear equations $H_j \in \mathbb{E}[x_0, \ldots, x_d]$ for $1 \leq j \leq d-1$, *i.e.*, E is an ordinary line in $\mathbb{P}^d_{\mathbb{E}}$. Then E is obviously rational and we again get $\overline{E}^2 = -1$.

Contracting Lines

Let $E \subset X$ be a contractible curve and $\phi : X \dashrightarrow X'$ the map induced by the blow down of E. Then ϕ is induced by the divisor $-\widetilde{K} + \overline{E} = \widetilde{D} + \overline{E}$ corresponding to $\omega^\vee_{X'}$.

Del Pezzos of Degree 1

In this case X' is a Del Pezzo surface of degree 2 and $\omega^\vee_{X'}$ is ample, but only its square is very ample. So we need the linear systems $\mathcal{L}_{\widetilde{X}}(\widetilde{D} + \overline{E})$ and $\mathcal{L}_{\widetilde{X}}(2(\widetilde{D} + \overline{E}))$ to compute the blow down of E.

Assume first that E comes in a pair, *i.e.*, $E = E_1$ and its complement is E_2. Then $L_m := \mathcal{L}_{\widetilde{X}}(m(\widetilde{D} + \overline{E})) \cong \mathcal{L}_X(mD + mE_1))$ for $m \in \{1, 2\}$. The equation $z - (ax^2 + bxy + cy^2)$ shows that $2D$ is linearly equivalent to $E_1 + E_2$. So we can as well compute the system $\mathcal{L}_X(3mD - mE_2)$. In other words we start with the system $\mathcal{L}_X(3mD)$ and restrict to those sections that pass through E_2 with multiplicity m.

Now assume that E is isolated. Then it passes through a singular point q. The equation $b_0x - a_0y$ shows that D, resp. $\widetilde{D}$, is linearly equivalent to E, resp. $\widetilde{E}$. We also know $\widetilde{E} = \overline{E} + R$. Then $L_m := \mathcal{L}_{\widetilde{X}}(m(\widetilde{D} + \overline{E})) \cong \mathcal{L}_{\widetilde{X}}(2m\widetilde{D} - mR)$ for $m \in \{1, 2\}$. In other words we start with the system $\mathcal{L}_{\widetilde{X}}(2m\widetilde{D})$ and restrict to those sections that pass through R with multiplicity m. But R contracts to q. So we can right away start with the system $\mathcal{L}_X(2mD)$ and restrict to those sections that pass through q with multiplicity m.

We construct a map to $\mathbb{P}_{\mathbb{E},(1,1,1,2)}$. So in both cases $\dim(L_1) = 3$, $\dim(L_2) = 7$ and the complement of L_1^2 in L_2 will be spanned by one element.

Del Pezzos of Degree 2

In this case X' is a Del Pezzo surface of degree 3 and $\omega^\vee_{X'}$ is very ample. So we only need the single linear system $\mathcal{L}_{\widetilde{X}}(\widetilde{D} + \overline{E})$ to compute the map.

We know that E comes in a pair, *i.e.*, $E = E_1$ and its complement is E_2. The equation $z - (ax + by)$ shows that D is linearly equivalent to $E_1 + E_2$. So we compute the system $L := \mathcal{L}_{\widetilde{X}}(2\widetilde{D} - \widetilde{E_2})$ instead. In other words we start with the system $\mathcal{L}_{\widetilde{X}}(2\widetilde{D})$ (resp. $\mathcal{L}_X(2D)$) and restrict to those sections that pass through $\widetilde{E_2}$ (resp. E_2). We construct a map to $\mathbb{P}^3_{\mathbb{E}}$, so we must have $\dim(L) = 4$.

Del Pezzos of Degree 3 and Higher

Let d be the degree of X. Then X' is a Del Pezzo surface of degree $d + 1$. We need the single linear system $\mathcal{L}_{\widetilde{X}}(\widetilde{D} + \overline{E})$ to compute the map.

Using the linear equation H_1, we can compute a curve R (not necessarily irreducible) s.t. D is linearly equivalent to $E + R$. More precisely, R is given by the colon ideal $J := (\langle H_1 \rangle + I) : \langle H_1, \ldots, H_{d-1} \rangle$. So we compute the system $L := \mathcal{L}_{\widetilde{X}}(2\widetilde{D} - \widetilde{R})$. In other words we start with the system $\mathcal{L}_{\widetilde{X}}(2\widetilde{D})$ (resp. $\mathcal{L}_X(2D)$) and restrict to those sections that pass through $\widetilde{R}$ (resp. vanish on J). We construct a map to $\mathbb{P}^{d+1}_{\mathbb{E}}$, so we must have $\dim(L) = d + 2$.

Implementation for Non-closed Fields

Now we translate the above results to an algebraically non-closed ground field $\mathbb{E}$, *i.e.*, we forget about our former restriction $\mathbb{E} = \overline{\mathbb{E}}$. Let $E \subset X$ be an irreducible curve and assume that $\overline{E} \times \operatorname{Spec} \overline{\mathbb{E}} = \bigcup_{1 \le j \le r} \overline{E}_j$ falls apart into r lines over the algebraic closure. Then E is contractible if and only if each of the lines $\overline{E}_j$ fulfills conditions 1) and 2b) and all the lines are disjoint.

We can do basically the same things as before to find and blow down several conjugate lines at once. But this time we will get only candidates for contractible lines because we do not check whether the corresponding lines over $\overline{\mathbb{E}}$ intersect.

In the remaining part we give algorithms for reducing Del Pezzo surfaces to *some* higher degree whenever possible over the ground field $\mathbb{E}$.

Del Pezzos of Degree 1

1. Compute $F|_{z=ax^2+bxy+cy^2} = w^2 - G$ and derive the condition ideal $I \subseteq \mathbb{E}[a, b, c]$ for G being a square (over $\overline{\mathbb{E}}$). Let $(a_i, b_i, c_i) \in \mathbb{E}_i^3$ for $1 \le i \le n$ be the roots of I where $\mathbb{E}_i$ is a residue field of degree $r_i := [\mathbb{E}_i : \mathbb{E}]$.

2. Let $J \subseteq \{1, \ldots, n\}$ be the set of indices s.t. for each $i \in J$ the polynomial $F|_{z=a_ix^2+b_ixy+c_iy^2}$ factors already over $\mathbb{E}_i$ and let $w - H_i(x, y)$ with $H_i \in \mathbb{E}_i[x, y]$ be a linear factor. Then the morphism

$$\mathbb{P}^1_{\mathbb{E}_i} \to \mathbb{P}_{\mathbb{E},(1,1,2,3)} : (x : y) \mapsto (x : y : a_ix^2 + b_ixy + c_iy^2 : H_i(x, y))$$

   parametrizes a *proper* curve $E_i \subset X$, actually corresponding to r_i distinct lines over the algebraic closure.

3. Let $(a_i : b_i : c_i)$ with $a_i, b_i, c_i \in \mathbb{E}_i$ for $n + 1 \le i \le m$ be the coordinates of the singular points of the curve defined by $F|_{w=0}$ (except $(0 : 0 : 1)$). We assume again that $\mathbb{E}_i$ is the residue field and let $r_i := [\mathbb{E}_i : \mathbb{E}]$. Then

$$\mathbb{P}^1_{\mathbb{E}_i} \to \mathbb{P}_{\mathbb{E},(1,1,2,3)} : (x : y) \mapsto (a_i : b_i : c_i : 0)$$

   has image a point $E_i \in X$ which we consider a *degenerate* curve. Set $J := J \cup \{n + 1, \ldots, m\}$.

4. If J is empty then X contains no contractible curves over $\mathbb{E}$, so report an error. Otherwise let $\iota \in J$ and set $J := J \setminus \{\iota\}$.

5. For $m \in \{1, 2\}$ compute the linear system L_m which is $\mathcal{L}_X(m(2+r_\iota-a)D)$ restricted to the sections that pass through E_ι with multiplicity m. Here $a = 0$ if E_ι is a proper curve and $a = 1$ if E_ι is a degenerate curve. Distinguish the following cases:

- If $r_\iota = 1$, $\dim(L_1) = 3$ (with basis, say, b_x, b_y, b_z) and the complement of L_1^2 in L_2 has dimension 1 (with generator, say, b_w) then the map $\pi : X \to \mathbb{P}_{\mathbb{E},(1,1,1,2)} : (x, y, z, w) \mapsto (b_x : b_y : b_z : b_w)$ is birational and has image a Del Pezzo X' of degree 2.
- If $2 \leq r_\iota \leq 8$ and $\dim(L_1) = r_\iota + 2$ (with basis, say, $b_0, \ldots, b_{r_\iota+1}$) then the map $\pi : X \to \mathbb{P}_{\mathbb{E}}^{r_\iota+1} : (x : y : z : w) \mapsto (b_0 : \cdots : b_{r_\iota+1})$ is birational and has image a Del Pezzo X' of degree $r_\iota + 1$.

If one of the above cases applies return π restricted to its image, otherwise go back to step 4 because E_ι is not contractible over $\mathbb{E}$.

It is obvious that we do not need to compute L_2 if $r_\iota \neq 1$. Further we would like to emphasize that we do not need to compute the implicit equations of the (proper or degenerate) curves E_i; Indeed, the system $\mathcal{L}_X(aD)$ is isomorphic to the forms in $\mathbb{E}[x, y, z, w]/\langle F \rangle$ of weighted degree a. The subspace of forms vanishing with order greater or equal to b on E_i can be computed by substituting the parametrization of E_i (no matter whether proper or degenerate) into the first $b - 1$ derivatives and reading of linear conditions. Finally using `TransformRelations` one computes a system of linear equations whose kernel determines $\mathcal{L}_X(aD - bE_i)$.

Del Pezzos of Degree 2

1. First we compute the $(a_i : b_i : c_i)$ with $a_i, b_i, c_i \in \mathbb{E}_i$ s.t. $F \bmod a_i x + b_i y + c_i z$ is reducible. We illustrate this for the case $c = 1$: Let $F_{z=-ax-bx} = w^2 - G$ and derive the condition ideal $I \subseteq \mathbb{E}[a, b]$ for G being a square (over $\overline{\mathbb{E}}$). Let $(a_i, b_i) \in \mathbb{E}_i^2$ be the roots of I where $\mathbb{E}_i$ is a residue field of degree $r_i := [\mathbb{E}_i : \mathbb{E}]$ (and $c_i = 1$). Then we proceed similarly for $c_i = 0$, $b_i = 1$ and $c_i = b_i = 0$, $a_i = 1$. Assume we have computed n triples (a_i, b_i, c_i) altogether.

2. Let $J \subseteq \{1, \ldots, n\}$ be the set of indices s.t. for each $i \in J$ the polynomial F modulo $a_i x + b_i y + c_i z$ factors already over $\mathbb{E}_i$ and let $w - H_i(x, y, z)$ with $H_i \in \mathbb{E}_i[x, y, z]$ be the representative of a linear factor. Further let $(X_i, Y_i, Z_i) \in \mathbb{E}_i[x, y, z]^3$ be a parametrization of the line $a_i x + b_i y + c_i z$. Then the morphism

$$\mathbb{P}^1_{\mathbb{E}_i} \to \mathbb{P}_{\mathbb{E},(1,1,2,3)} : (x : y) \mapsto (X_i : Y_i : Z_i : H_i)$$

parametrizes a curve $E_i \subset X$, actually corresponding to r_i distinct lines over the algebraic closure.

3. If J is empty then X contains no contractible curves over $\mathbb{E}$, so report an error. Otherwise let $\iota \in J$ and set $J := J \setminus \{\iota\}$.

4. Compute the linear system L which is $\mathcal{L}_X(2D)$ restricted to the sections that pass through E_ι.

5. If $1 \leq r_\iota \leq 7$ and $\dim(L) = r_\iota + 3$ (with basis, say, $b_0, \ldots, b_{r_\iota+2}$) then the map $\pi : X \to \mathbb{P}_{\mathbb{E}}^{r_\iota+2} : (x : y : z : w) \mapsto (b_0 : \cdots : b_{r_\iota+2})$ is birational and has image a Del Pezzo X' of degree $r_\iota + 2$. Return π restricted to its image, otherwise go back to step 3 because E_ι is not contractible over $\mathbb{E}$.

Del Pezzos of Degree 3 and Higher

1. First compute the vanishing ideals $V_i \subset \mathbb{E}[x_0, \ldots, x_d]$ (over the ground field) of all "lines" $E_i \subset X$ for $1 \leq i \leq n$ and set $J := \{1, \ldots, n\}$. A line here means again, that E_i is parametrized linearly by some $\mathbb{P}^1_{\mathbb{E}_i}$ and corresponds to $r_i := [\mathbb{E}_i : \mathbb{E}]$ distinct lines over the algebraic closure.

2. If J is empty then X contains no contractible curves over $\mathbb{E}$, so report an error. Otherwise let $\iota \in J$ and set $J := J \setminus \{\iota\}$.

3. Choose a form $G \in V_\iota$ of *low* degree e vanishing on E_i with multiplicity one. Compute $J := (\langle G \rangle + I) : V_\iota \subset \mathbb{E}[x_0, \ldots, x_d]$.

4. Compute the linear system L which is $\mathcal{L}_X((e+1)D)$ restricted to the sections that vanish on J.

5. If $1 \leq r_\iota \leq 9 - d$ and $\dim(L) = d + r_\iota + 1$ (with basis, say, $b_0, \ldots, b_{d+r_\iota}$) then the map $\pi : X \to \mathbb{P}^{d+r_\iota}_{\mathbb{E}} : (x_0 : \cdots : x_d) \mapsto (b_0 : \cdots : b_{d+r_\iota})$ is birational and has image a Del Pezzo X' of degree $d + r_\iota$. In this case return π restricted to its image, otherwise go back to step 2 because E_ι is not contractible over $\mathbb{E}$.

Note that multiples of G are for sure contained in the linear system L. So we can assume w.l.o.g. that $b_i = x_i G$ for $0 \leq i \leq d$. In this case the inverse of π (restricted to its image) is just the projection to the first coordinates $(b_0 : \cdots : b_{d+r_\iota}) \mapsto (b_0 : \cdots : b_d)$ and comes for free.