DOI: 10.1478/AAPP.942A7

AAPP | Atti della Accademia Peloritana dei Pericolanti
Classe di Scienze Fisiche, Matematiche e Naturali
ISSN 1825-1242

Vol. 94, No. 2, A7 (2016)

A COMPUTATIONAL APPROACH TO THE THEORY OF ADJOINTS

MICHELA CERIA[*]

(communicated by Paolo Valabrega)

ABSTRACT. The word "adjoint" refers to several definitions which are not all equivalent (see, for example, Greco and Valabrega 1979, 1982): we shall deal with any of them. The aim of this work is to provide an algorithm which, given two plane curves D, H, allows one to decide whether H is adjoint to D. With a slight modification to the main procedure, we shall be able to deal with special adjoints (Greco and Valabrega 1979, 1982) and true adjoints (Oneto 1979).

1. Introduction

The notion of adjoint curve to a given plane curve is a classical one, and goes back to Brill and Nöether (1874). Roughly speaking, a curve H, to be adjoint to a singular curve D, is required to pass with multiplicity at least $r-1$ through each r-th point of D. Almost eighty years later, basing on Zariski's ideas, Gorenstein introduced a new definition of adjoint, related to the conductor sheaf of the given curve. Greco and Valabrega (1979, 1982) investigated the different definitions, appearing in literature, and the relationships among them.

In this paper we investigate the theory of adjoints to a plane curve from a computational point of view. More precisely, we provide an algorithm in order to decide whether, given two plane curves D, H, H is adjoint to D according to the several (not necessarily equivalent) definitions

2. A brief summary of adjoint theory

In this paragraph we recall the seven definitions of adjoint curve given by Greco and Valabrega (1979, 1982) and refer to the related papers for all the details and proofs.

In what follows curves will be denoted by capital letters $(C, H, D...)$; moreover, a reduced and irreducible plane curve is called *integral plane curve*.

If a plane curve D has some singular points, we can perform a chain of blow up's along them, in order to desingularize it. An *actual point* of D is a point P lying on the support of D, a *neighbouring point* of an actual point P is a point Q of a D-scheme $f : Z \to D$, obtained by a finite sequence of blowing up's centered at a closed point, such that $f(Q) = P$.

Definition 1 (Brill-Noether)**.** *A plane curve H is* adjoint *(shortly A1) to an integral plane curve D if it passes through every point P of multiplicity r (actual or neighbouring) with multiplicity at least $r-1$ (see Brill and Nöether 1874; Greco and Valabrega 1979).*

Since the concepts of *divisor of the plane* and of *curve* coincide, sometimes the adjoint curves are also called *adjoint divisors* and we will denote them with capital letters, by abuse of notation (Greco and Valabrega 1979).

Definition 2 (Gorenstein)**.** *A plane curve H is* adjoint *(A2) to an integral plane curve D if it belongs to the conductor of D (see Gorenstein 1952).*

Definition 3 (Keller)**.** *A plane curve H is adjoint (A3) to an integral plane curve D if it cuts out the divisor of double points of D (see Keller 1974; Greco and Valabrega 1979, Definitions 2.13, 4.3).*

Definition 4. *A plane curve H is a* special adjoint *to an integral plane curve D (SA) if it has a point of multiplicity exactly $r-1$ at every singular point having multiplicity r in D, actual or infinitely near (see Greco and Valabrega 1979)).*

Now we list various relations among the different notions of adjoint, referring to Greco and Valabrega (1979) for proofs.

Theorem 5 (Greco and Valabrega 1979, 4.6)**.** *A curve H is A2 to some D if and only if it is A3.*

Theorem 6 (Greco and Valabrega 1979, 4.13)**.** *Let H be a plane curve and D another plane curve, that is not a component of H. Then:*

(1) *if H is A1 to D, then it is also A2;*
(2) *if H is A2 to D, then there exist:*
   - *a curve Z not passing through the singular points of D or the common points of H and D;*
   - *t plane curves $H_1,\dots,H_t$ of the same degree d which are SA to D, such that the divisor cut by H on D coincides, except possibly on Z, with the divisor cut on D by a suitable element of the linear system generated by the curves $H_1,\dots,H_t$.*

Corollary 7 (Greco and Valabrega 1979, 4.16)**.** *Suppose that all the singular points of D are actual. Then the curve H is A1 if and only if it is A2 if and only if it is A3.*

Proposition 8 (Greco and Valabrega 1979, 5.1)**.** *Let $\mathscr{L}$ be a linear system of plane curves, such that D is not a fixed component. If the generic element of L is A2 or A3 to D, then all the elements of $\mathscr{L}$ are so.*

Lemma 9 (Greco and Valabrega 1979, 5.3)**.** *Let H_1,H_2 be curves generating linearly equivalent divisors $div(H_i)=\sum_{P\in X_i} ord_P(H_i)P$, $i=1,2$, where X_i is the nonsingular model of H_i, having at the point P the same multiplicity s. Let $H_i^{\prime}$ be the two strict transforms obtained blowing up in P, then such transforms form linearly equivalent divisors.*

Proposition 10 (Greco and Valabrega 1979, 5.4)**.** *Let $\mathscr{L}$ be a linear system of plane curves, such that its elements are A2 for the curve D. If $\mathscr{L}$ contains at least an adjoint SA, then the generic element of this system is SA.*

Corollary 11 (Greco and Valabrega 1979, 5.5). *Let D be a projective plane curve and $\mathscr{L}_n$ be the linear system of its A2 curves of degree n. If $n \gg 0$, the generic element of $\mathscr{L}_n$ is SA.*

If the curves of lemma **??** are not adjoint SA, then lemma **??** does not hold. For the adjoints A1, proposition **??** is false.

Now we want to introduce two other definitions of adjoint (A4,A5). The first one is strictly related to the concept of *virtual multiplicity* and it was developed by italian algebraic geometers.

Definition 12. *A curve D passes through a "point" P with* virtual multiplicity *s, if $r \geq s$, where r is the (effective) multiplicity of D at P.*

The second definition was studied for the first time by Abhyankar and Sathaye (1974).

Definition 13. *The effective divisor D of the plane is A4 for a curve C if and only if D passes through all the singular points of C having multiplicity r, actual or infinitely near, with virtual multiplicity $r-1$.*

Definition 14. *The effective divisor $H \subset X$ is A5 for a curve D if and only if*

$$H \geq \sum_{i=0}^{n} (r_i - 1) P_i,$$

where the P_i are the singular points of D, actual or infinitely near, and r_i are their multiplicities on D.

Theorem 15 (Greco and Valabrega 1982, 2.3). *Let H be an effective divisor of the plane. Then the following facts are equivalent:*

(1) *H is A2;*
(2) *H is A4;*
(3) *H is A5.*

Mnuk (1997) proves that, for the case of absolutely irreducible plane curves, it also holds

$$H \text{ is A1} \Leftrightarrow H \text{ is A2}.$$

but this is not true in general.

Now we define the concept of *true adjoint* to a curve D (see Oneto 1979).

Definition 16. *H is a* true adjoint *to D if, for every point $Q \in D$, its local equation in Q generates in $\overline{\mathscr{O}_Q(D)}$ the conductor $\mathscr{C}_{\overline{\mathscr{O}_Q(D)}/\mathscr{O}_Q(D)}$.*

If C is an affine plane curve, we obtain the definition given by Abhyankar and Sathaye (1974):

Definition 17. *A* true adjoint *to C is an effective divisor of the affine plane such that, if $h \in k[X,Y]$ is one of its equations, $\overline{h}$ its image in the coordinate ring of C, then $\overline{h}$ generates the conductor in $\overline{\Gamma[C]}$.*

3. Computational classification of adjoint curves

In this paragraph we shall develop an algorithm in order to answer the following question:

Given two plane curves D, H, is it possible to estabilish whether H is adjoint to D at least according to some of the seven given definitions?

We shall need to desingularize the two given curves simultaneously, construct the conductor ideal and then decide whether H is adjoint to D.

In order to classify adjoints A2, A3, A4, A5, whose definitions are equivalent, we could also choose different strategies instead of constructing the conductor ideal. Indeed we are considering the definition of adjoint A2, but we could change perspective and work with A3, A4 or A5.

We chose to construct the conductor for three main reasons:

- knowing the conductor we can deal with adjoints AV (through localization), so we do not have to set up a specific computation for them;
- the conductor also allows to *construct* all the adjoints A2;
- it is possible to construct the conductor exploiting a part of the computation already done for dealing with adjoints A1 and AV, i.e., the blowing up process.

An approach for the case of Gorenstein adjoints (A2) has been proposed by El Kahoui and Moussa (2014), using Groebner bases techniques.

In this paper, given an irreducible curve C, defined by a polynomial $f \in k[x,y]$ and denoted by $\pi : k[x,y] \to k[C]$ the canonical homomorphism, where $k[C]$ is the coordinate ring of C, a lexicographical Groebner basis of the preimage w.r.t. π of the conductor ideal is computed. Our algorithm is an alternative to that of El Kahoui and Moussa (2014), which does not rely on the use of Groebner bases. Moreover, not only the conductor is computed, but we can also test whether a curve H is adjoint to D according to Brill-Noether definition and we study special adjoints, whose definition is strictly connected to the one of adjoints A1.

Note that an affine algebraic set V is irriducibile if and only if $I(V)$ is a prime ideal.

The first step is to find all the singular points of the given curve i.e., the points at which to perform the blowing up process. Such result can be easily achieved through the "Jacobian Test":

Definition 18. *Let $X \subseteq \mathbb{A}^n(k)$ be an affine variety, $k = \overline{k}$ an algebraically closed field. Let $\{f_1, .., f_s\} \subset A = k[X_1, ..., X_n]$ be a generators' set of the ideal $I(X)$ of X. The variety X is called* nonsingular at a point $P \in X$ *if the* Jacobian Matrix

$$J_P = \left(\frac{\partial f_i}{\partial x_j}(P)\right)$$

has $rank(J_P) = n - r$, where $r = dim(X)$. If this condition holds at every point, X is a nonsingular variety. *X is called* singular *if there exists at least a point P such that $rank(J_P) < n - r$, i.e., a* singular point.

We then use this definition to find the singular points of the given irreducible curve D. More precisely, we construct the Jacobian matrix J and then we solve the polynomial equations' system formed by:

- the equation defining D;
- the 2×2 minors of J: in fact, we look for the points P such that $rank(J_P) = 1$.

Such a problem can be solved with the software Singular (Decker *et al.* 2015), namely using the libraries `primdec.lib`, to compute all the minors of J and `solve.lib` to solve the obtained equation system. The software Singular has some pre-implemented procedures to obtain this result and also to compute the multiplicity of each singular point of D.

Example 19. *Take the plane curve D, defined by the equation $y^2 - x^3 - x^2$. Performing a simple computation we will obtain its singularities:*

```
> LIB"primdec.lib";
> radical(I);
_[1]=x2-y2+x3
> list l=minAssGTZ(I);
> l;
[1]:
_[1]=x2-y2+x3
> ideal K=l[1];
> std(K);
_[1]=x2-y2+x3
> ideal sing=l[1]+minor(jacob(l[1]),1);
> std(sing);
_[1]=x
_[2]=y
```

We obtained this way that the singularity locus is $I = (x,y)$, representing the origin. If we also type

```
> LIB "solve.lib";
> solve(sing);
```

we obtain the coordinate of the desired point:

```
[1]:
[1]:
0
[2]:
0.
```

We then compute its multiplicity:

```
> mult(std(I));
2.
```

Actually, the origin is a double point for our curve D and it is the only singular point.

When the set $\mathbf{Y} := \{(a_{1,1}, a_{1,2}), ..., (a_{s,1}, a_{s,2})\}$ of the singular points of D is known, we have to blow up both D and H at its elements. Our blowing up algorithm is based on the techniques explained by Fulton (1989).

First of all we generate a list containing the $s = |\mathbf{Y}|$ matrices of the following shape:

$$M_1 = \begin{bmatrix} x_1 - a_{1,1} & x_2 - a_{1,2} \\ x_3 & x_4 \end{bmatrix}$$

$$\vdots$$

$$M_s = \begin{bmatrix} x_1 - a_{s,1} & x_2 - a_{s,2} \\ x_{2s+1} & x_{2(s+1)} \end{bmatrix},$$

for each one of the s points in $\mathbf{Y}$ we define 2 new variables (i.e., $x_3, ..., x_{2(s+1)}$). In order to construct all the possible affine charts where to blow up, we perform the following steps:

(1) consider the matrix M_1
  - create a copy $C_{1,1}$, of it substituting x_3 with 1;
  - store $C_{1,1}$ in a list L_1;
  - create a copy $C_{1,2}$ of it substituting x_4 with 1;
  - store $C_{1,2}$ in a list L_1;

(2) do the same with $M_2, ..., M_s$.

Afterwards, we take in all the possible ways an element from each list $L_1, ..., L_s$ obtained by the previous procedure, obtaining $h = s^2$ lists $N_1, ..., N_h$. They represent all the possible affine charts where to blow up our curves. Indeed, the equations of each chart can be obtained computing the 2×2 determinants of the matrices of every such list. This computations can generate "superfluous expressions", i.e., polynomials contained in the ideal generated by the other ones. We then eliminate them, obtaining a set of generators for each affine chart without any *interdependence*. We use now such determinants and the matrices representing the current affine chart to compute the exceptional divisors (one for each point). More precisely we will solve the systems formed by:

- the obtained non-superfluous determinants;
- the first row of each matrix,

taking all the matrices one by one.

The next step consists of finding the strict transform, so we have to eliminate the exceptional divisors from the algebraic variety represented by the ideal generated by

(1) the determinants relative to the affine chart;
(2) the equation of the curve.

We perform such an "elimination" through successive division of ideals, looking for the strict transform. Indeed, we point out that if $I, J \triangleleft k[X_1, ..., X_n]$ are two ideals (Fulton 1989; Mora 2005), we have

$$V(I : J) \supseteq \overline{V(I) \setminus V(J)}.$$

If $k = \overline{k}$ and I is a radical ideal then

$$V(I : J) = \overline{V(I) \setminus V(J)}.$$

The neighbouring points can be easily found solving the system formed by the exceptional divisors and the strict transform. If such system has no solutions this means that we chose the wrong affine chart (see Fulton 1989) and we must repeat the process with another affine chart. Since there are a finite number, we reach the correct result in finitely many steps. Using the Jacobian test, we detect the singular points and then again their multiplicitiy.

Repeating on the strict transform all the steps above till the infinitely near points are smooth we completely desingularize the curve. Since the blowing up process ends in finitely many steps, also our procedure does.

There are several ways to compute the conductor ideal (see, for example, Boehm *et al.* 2015). We develop a simple construction which is related to the above blowing up process. Only small modifications to the original blow up process are needed. In order to compute the conductor ideal we only have to multiply the ideals of blowing up centers (i.e., the singular points of the given curve and the neighbouring points) at the power "multiplicity minus one". A simple control on the localization of the conductor allows us to detect true adjoints. In order to classify adjoints A1 or SA we again modify the blowing up process. Consider two curves C, C' such that C' is a candidate to be adjoint to C. We first check whether C' belongs to the conductor of C. If not, such a curve is neither A1 nor SA. Instead, if it is so, we check multiplicities of C, C' at the singular points of C, using the definition of A1 or SA stated in the previous paragraph. If they do not fail the test, we start blowing up both curves and testing the multiplicities of the neighbouring points of C.

4. The procedures

We now provide a pseudocode for the procedures developing the steps explained in the previous paragraph. The procedure MatrixConstr computes the s matrices necessary to construct the affine charts.

```
1: procedure MATRIXCONSTR($V, A, P$) $\to M$    ▷ MatrixConstr takes variables and points in input, in order to construct the matrices introduced before.
Require:
2: $V$: list containing *all* the used variables;
3: $A$: list of variables obtained by $V$ removing all the elements imposed to be equal to 1 in the previous step;
4: $P$: list containing all points.
5:     $A1 = A$
6:     $V1 = V$    ▷ These are copies of the lists $A$ and $V$ given as input.
7:     $b = size(A)$
8:     $M = [\,]$
9:     for $i = 1$ to $|P|$ do
10:        $c =$ maximal index of $A1$
11:        $INDNEW = [seq(x[j], j = c+1, ..., c+b)]$    ▷ These are the indexes of the new variables.
12:        $Ma = Matrix(2, b, [[seq(A[k] - P[i][k], k = 1..size(A))], INDNEW])$    ▷ These are the matrices which will be inserted in the list $M$.;
13:        $M = [M, Ma]$    ▷ Updates the list $M$.;
14:        $A1 = [A1, INDNEW]$    ▷ Updates the list $A1$.
15:        $V1 = [V1, INDNEW]$    ▷ Updates the list $V1$.
16:    end for
       return $M$
17: end procedure
```

```
1: procedure CHARTCONSTR($M,o,v$) $\to H$ ▷ CharConstr takes the matrices contained in $M$ and it constructs all the affine charts where we can blow up.
Require:  • $M$: list of matrices obtained using MatrixConstr;
          • $o$: number of $A$'s elements;
          • $v$: maximal index of the elements contained in $A$.
2:      $n = size(M)$;
3:      $l = no$;
4:      $p = 1$;
5:      for $i = 1$ to $n$ do
6:          $p = po$
7:      end for
8:      $t = pl$;
9:      $S = Vector[row](t)$;
10:     $g = t$;
11:     for $j = 0$ by $o$ to $l-1$ do    ▷ Be careful to this step: $j$ is incremented by $o$, nor by 1 as usually is!
12:         $g = \frac{g}{o}$;
13:         for $k = j$ by $l$ to $t-1$ do    ▷ $k$ is incremented by $l$.
14:             $r = k - (mod(k,g))$;
15:             for $q = 0$ to $o-1$ do
16:                 if $mod(floor(r/g),o) == q$ then    ▷ Floor is the maximal integer $f \leq \frac{r}{g}$.
17:                     $S[k+q+1] = 1$
18:                 end if
19:             end for
20:         end for
21:     end for
22:     $H = [\,]$;
23:     for $h = 0$ to $p-1$ do
24:         $N = M$;
25:         for $d = 0$ to $l-1$ do
26:             $y = hl + d + 1$;
27:             if $S[y] = 1$ then
28:                 $N = subs(x[d+v+1] = 1, N)$    ▷ The variable $x[d+v+1]$ is imposed equal to 1.
29:             end if
30:         end for
31:         $H = [H,N]$
32:     end for
        return $H$
33: end procedure
```

```
1: procedure DETERMINANTS2X2GEN($M$) $\to H$ ▷ Determinants2X2gen computes the necessary determinants.
2:     $det = determinantsmin(deletecol(M, ColumnDimension(M[1]) - 2))$ ▷ we produce all the $2 \times 2$ determinants deleting the suitable columns.
       return $det$.
3: end procedure
```

```
1: procedure SUBSSTGEN($Z$) $\to B$ ▷ subsstgen eliminates the superfluous determinants from their list $Z$.
2:     $A = Z$;
3:     $B = []$;
4:     for $i = 1$ to $|A|$ do
5:         if not $IdealMembership(A[i], PolynomialIdeal(op(subsop(i = NULL, A))))$ then
6:             $B = Flatten([B, A[i]])$
7:         end if
8:     end for
       return $B$
9: end procedure
```

```
1: procedure ECDIV($CHARTS, S$) $\to L$ ▷ ecdiv computes the exceptional divisors.
2:     $L = [\,]$;
3:     for $i = 1$ to $|CHARTS|$ do
4:         $B = Row(CHARTS[i], 1)$ ▷ “Row" means extracting a row from the given matrix.
5:         $C = convert(B, list)$;
6:         $G = solve([C, S])$;
7:         if $G == NULL$ then
8:             $G = [\,]$
9:         end if;
10:        $K = convert(G, list)$;
11:        $M = [\,]$;
12:        for $j = 1$ to $|K|$ do
13:            $M = [M, lhs(K[j]) - rhs(K[j])]$ ▷ rhs and lhs are the right and the left member of an equation.
14:        end for
15:        $J = (M)$
16:        $L = [L, J]$
17:    end for
       return $L$
18: end procedure
```

```
1: procedure ELIMECDIV($H,L$) $\to J$          ▷ elimecdiv computes the strict transform.
2:     $J = H$;
3:     $J1 = H$;
4:     $a = false$;
5:     $Id = (1)$;
6:     for $i = 1$ to $|L|$ do
7:         Id $= Id \cap L[i]$
8:
9:     end for
10:    while not $a$ do
11:        $J = J : Id$;
12:        if $J \subseteq J1$ and $J1 \subseteq J$ then
13:            $a = true$;
14:            $J1 = J1 : Id$
15:        else
16:            $J1 = J1 : Id$
17:        end if
18:    end while
         return $J$
19: end procedure
```

5. Some examples

Consider the following curve, presenting a singularity called *tacnode*:

Tac;

$$[x_2^2 - x_1^4 - x_2^4]$$

$M = MatrixConstr([x_1,x_2],[x_1,x_2],[[0,0]]);$

$$\left[\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}\right]$$

$CHARTS = ChartConstr(M,2,2);$

$$\left[\begin{bmatrix} x_1 & x_2 \\ 1 & x_4 \end{bmatrix}, \begin{bmatrix} x_1 & x_2 \\ x_3 & 1 \end{bmatrix}\right]$$

$CHARTS = LengthSplit(C,1);$

$$\left[\left[\begin{bmatrix} x_1 & x_2 \\ 1 & x_4 \end{bmatrix}\right], \left[\begin{bmatrix} x_1 & x_2 \\ x_3 & 1 \end{bmatrix}\right]\right]$$

$CHARTS[1];$

$$\left[\begin{bmatrix} x_1 & x_2 \\ 1 & x_4 \end{bmatrix}\right]$$

$DetS = Determinants2X2gen(CHARTS[1]);$

$$[x_1x_4 - x_2]$$

$L1 = ecdiv(CHARTS[1], DetS);$

$$[(x_1, x_2)]$$

$L2 = elimecdiv((op(DetS), op(Tac)), L1);$

$$(x_1x_4 - x_2, -x_2x_4 + x_1^3 + x_2^3x_4, -x_4^2 + x_1^2 + x_2^2x_4^2, -x_2^2 + x_1^4 + x_2^4)$$

$solve([x_1, x_2, x_1x_4 - x_2, -x_2x_4 + x_1^3 + x_2^3x_4, -x_4^2 + x_1^2 + x_2^2x_4^2, -x_2^2 + x_1^4 + x_2^4]);$

$$\{x_4 = 0, x_1 = 0, x_2 = 0\}, \{x_4 = 0, x_1 = 0, x_2 = 0\}$$

Using Singular, we obtain that $[0,0,0]$ is the only double point of the strict transform:

$M1 = MatrixConstr([x_1, x_2, x_3, x_4], [x_1, x_2, x_4], [[0,0,0]]);$

$$\left[\begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & x_6 & x_7 \end{bmatrix}\right]$$

$CHART2 = ChartConstr(M1, 3, 4);$

$$\left[\begin{bmatrix} x_1 & x_2 & x_4 \\ 1 & x_6 & x_7 \end{bmatrix}, \begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & 1 & x_7 \end{bmatrix}, \begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & x_6 & 1 \end{bmatrix}\right]$$

$CHARTS2 = LengthSplit(CHART2, 1);$

$$\left[\left[\begin{bmatrix} x_1 & x_2 & x_4 \\ 1 & x_6 & x_7 \end{bmatrix}\right], \left[\begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & 1 & x_7 \end{bmatrix}\right], \left[\begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & x_6 & 1 \end{bmatrix}\right]\right]$$

$CHARTS2[1];$

$$\left[\begin{bmatrix} x_1 & x_2 & x_4 \\ 1 & x_6 & x_7 \end{bmatrix}\right]$$

$DetS2 = Determinants2X2gen(CHARTS2[1]);$

$$[x_2x_7 - x_4x_6, x_1x_7 - x_4, x_1x_6 - x_2]$$

$S = subsstgen(DetS2);$

$$[x_1x_7 - x_4, x_1x_6 - x_2]$$

$L1 = ecdiv(CHARTS2[1], S);$

$$[(x_1, x_2, x_4)]$$

$L2 = elimecdiv((x_1x_7 - x_4, x_1x_6 - x_2, x_1x_4 - x_2, -x_2x_4 + x_1^3 + x_2^3x_4, -x_4^2 + x_1^2 + x_2^2x_4^2, -x_2^2 + x_1^4 + x_2^4), L1);$

$$(-x_2x_7 + x_4^2, 1 - x_7^2 + x_7^2x_2^2, x_1x_4 - x_2, -x_4 + x_6, x_1x_7 - x_4,$$
$$-x_2x_4 + x_1^3 + x_2^3x_4, -x_4x_7 + x_1 + x_2^2x_4x_7, x_1^2 - x_2x_7 + x_7x_2^3, -x_2^2 + x_1^4 + x_2^4)$$

$solve([x_1, x_2, x_4, -x_2x_7 + x_4^2, 1 - x_7^2 + x_7^2x_2^2, x_1x_4 - x_2, -x_4 + x_6, x_1x_7 - x_4,$
$-x_2x_4 + x_1^3 + x_2^3x_4, -x_4x_7 + x_1 + x_2^2x_4x_7, x_1^2 - x_2x_7 + x_7x_2^3, -x_2^2 + x_1^4 + x_2^4]);$

$$\{x_4 = 0, x_1 = 0, x_2 = 0, x_6 = 0, x_7 = 1\},\ \{x_4 = 0, x_1 = 0, x_2 = 0, x_6 = 0, x_7 = -1\}$$

So, if we want to compute the conductor we multiply $(x_1, x_2)(x_1, x_2, x_4)$ and, considering that $x_2 = x_1x_4$, we obtain (x_1^2, x_2).

We now show that x_2 is A1 and SA to the tacnode.

Tac;

$$[x_2^2 - x_1^4 - x_2^4]$$

$controllomoltA1([2],[1]);$

$$true$$

$controllomoltAS([2],[1]);$

$$true$$

$M = MatrixConstr([x_1,x_2],[x_1,x_2],[[0,0]]);$

$$\left[\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}\right]$$

$C = ChartConstr(M,2,2);$

$$\left[\begin{bmatrix} x_1 & x_2 \\ 1 & x_4 \end{bmatrix}, \begin{bmatrix} x_1 & x_2 \\ x_3 & 1 \end{bmatrix}\right]$$

$CHARTS = LengthSplit(C,1);$

$$\left[\left[\begin{bmatrix} x_1 & x_2 \\ 1 & x_4 \end{bmatrix}\right], \left[\begin{bmatrix} x_1 & x_2 \\ x_3 & 1 \end{bmatrix}\right]\right]$$

$CHARTS[1];$

$$\left[\begin{bmatrix} x_1 & x_2 \\ 1 & x_4 \end{bmatrix}\right]$$

$DetS = Determinants2X2gen(CHARTS[1]);$

$$[x_1x_4 - x_2]$$

$L1 = ecdiv(CHARTS[1],DetS);$

$$[(x_1,x_2)]$$

$L2 = elimecdiv((elementi\, di\, DetS, elementi\, di\, Tac),L1);$

$$(x_1x_4 - x_2, -x_2x_4 + x_1^3 + x_2^3x_4, -x_4^2 + x_1^2 + x_2^2x_4^2, -x_2^2 + x_1^4 + x_2^4)$$

$L3 = elimecdiv((elementi\, di\, DetS, x[2]),L1)$

$$(x_2,x_4)$$

$solve([x_1,x_2,x_1x_4 - x_2, -x_2x_4 + x_1^3 + x_2^3x_4, -x_4^2 + x_1^2 + x_2^2x_4^2, -x_2^2 + x_1^4 + x_2^4]);$

$$\{x_4 = 0, x_1 = 0, x_2 = 0\}, \{x_4 = 0, x_1 = 0, x_2 = 0\}$$

$[0,0,0]$ is the only double point of the strict transform. After checking multiplicities, we blow up again:

$M1 = MatrixConstr([x_1,x_2,x_3,x_4],[x_1,x_2,x_4],[[0,0,0]]);$

$$\left[\begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & x_6 & x_7 \end{bmatrix}\right]$$

$CHART2 = ChartConstr(M1,3,4);$

$$\left[\begin{bmatrix} x_1 & x_2 & x_4 \\ 1 & x_6 & x_7 \end{bmatrix}, \begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & 1 & x_7 \end{bmatrix}, \begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & x_6 & 1 \end{bmatrix}\right]$$

$CHARTS2 = LengthSplit(CHART2, 1);$

$$\left[\left[\begin{bmatrix} x_1 & x_2 & x_4 \\ 1 & x_6 & x_7 \end{bmatrix}\right], \left[\begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & 1 & x_7 \end{bmatrix}\right], \left[\begin{bmatrix} x_1 & x_2 & x_4 \\ x_5 & x_6 & 1 \end{bmatrix}\right]\right]$$

$CHARTS2[1];$

$$\left[\begin{bmatrix} x_1 & x_2 & x_4 \\ 1 & x_6 & x_7 \end{bmatrix}\right]$$

$DetS2 = Determinants2X2gen(CHARTS2[1]);$

$$[x_2x_7 - x_4x_6, x_1x_7 - x_4, x_1x_6 - x_2]$$

$S = subsstgen(DetS2);$

$$[x_1x_7 - x_4, x_1x_6 - x_2]$$

$L1 = ecdiv(CHARTS2[1], S);$

$$[(x_1, x_2, x_4)]$$

$L2 = elimecdiv((x_1x_7 - x_4, x_1x_6 - x_2, x_1x_4 - x_2, -x_2x_4 + x_1^3 + x_2^3x_4, -x_4^2 + x_1^2 + x_2^2x_4^2, -x_2^2 + x_1^4 + x_2^4), L1);$

$$(-x_2x_7 + x_4^2, 1 - x_7^2 + x_7^2x_2^2, x_1x_4 - x_2, -x_4 + x_6, x_1x_7 - x_4,$$
$$-x_2x_4 + x_1^3 + x_2^3x_4, -x_4x_7 + x_1 + x_2^2x_4x_7, x_1^2 - x_2x_7 + x_7x_2^3, -x_2^2 + x_1^4 + x_2^4)$$

$L3 = elimecdiv((elementi\ di\ S, x_2, x_4), L1)$

$$(x_2, x_4, x_6, x_7)$$

$solve([x_1, x_2, x_4, -x_2x_7 + x_4^2, 1 - x_7^2 + x_7^2x_2^2, x_1x_4 - x_2, -x_4 + x_6, x_1x_7 - x_4,$
$-x_2x_4 + x_1^3 + x_2^3x_4, -x_4x_7 + x_1 + x_2^2x_4x_7, x_1^2 - x_2x_7 + x_7x_2^3, -x_2^2 + x_1^4 + x_2^4]);$

$$\{x_4 = 0, x_1 = 0, x_2 = 0, x_6 = 0, x_7 = 1\},\ \{x_4 = 0, x_1 = 0, x_2 = 0, x_6 = 0, x_7 = -1\}$$

x_2 is A1 and SA to the tacnode.

Acknowledgments

I would like to offer my special thanks to Paolo Valabrega and Margherita Roggero for their great help while writing this work. Advice given by the referees has been a great help in the revision phase.

References

Abhyankar, S. S. and Sathaye, A. M. (1974). *Geometric theory of algebraic space curves*. Lecture Notes in Mathematics. Springer. URL: http://www.springer.com/us/book/9783540069690.

Boehm, J., Decker, W., Laplagne, S., and Pfister, G. (2015). "Local to global algorithms for the Gorenstein adjoint ideal of a curve". arXiv: 1505.05040.

Brill, A. and Nöether, M. (1874). "Ueber die algebraischen Functionen und ihre Anwendung in der Geometrie". *Mathematische Annalen* **7**(2), 269–310. DOI: 10.1007/BF02104804.

Decker, W., Greuel, G.-M., Pfister, G., and Schönemann, H. (2015). SINGULAR, V. 4-0-2 – *A computer algebra system for polynomial computations*. URL: http://www.singular.uni-kl.de.

El Kahoui, M. and Moussa, Z. Y. (2014). "An Algorithm to Compute the Adjoint Ideal of an Affine Plane Algebraic Curve". *Mathematics in Computer Science* **8**(2), 289–298. DOI: 10.1007/s11786-014-0193-x.

Fulton, W. (1989). *Algebraic Curves. An Introduction to Algebraic Geometry*. Advanced Book Classics. Addison-Wesley. URL: http://www.math.lsa.umich.edu/~wfulton/.

Gorenstein, D. (1952). "An arithmetic theory of adjoint plane curves". *Transactions of the American Mathematical Society* **72**(3), 414–436. URL: http://www.ams.org/journals/tran/1952-072-03/S0002-9947-1952-0049591-8/S0002-9947-1952-0049591-8.pdf.

Greco, S. and Valabrega, P. (1979). "On the theory of adjoints". In: *Algebraic Geometry: Summer Meeting, Copenhagen, August 7–12, 1978*. Ed. by K. Lønsted. Vol. 732. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer, pp. 98–123. DOI: 10.1007/BFb0066640.

Greco, S. and Valabrega, P. (1982). "On the theory of adjoints II". *Rendiconti del Circolo Matematico di Palermo* **31**(1), 5–15. DOI: 10.1007/BF02849535.

Keller, O. H. (1974). *Vorlesungen über algebraische Geometrie*. Vol. 71 Abb. Leipzig: Akademische Verlagsgesellschaft Geest & Portig K.-G. DOI: 10.1002/zamm.19750550927.

Mnuk, M. (1997). "An algebraic approach to computing adjoint curves". *Journal of Symbolic Computation* **23**(2-3), 229–240. DOI: 10.1006/jsco.1996.0085.

Mora, T. (2005). *Solving polynomial equation systems II: Macaulay's paradigm and Gröbner technology*. Cambridge: Cambridge University Press.

Oneto, A. (1979). "Conduttore e vere aggiunte ad una curva su una superficie". *Rendiconti del Seminario Matematico del Politecnico di Torino* **37**, 159–171.

* Università degli Studi di Trento
Dipartimento di Ingegneria e Scienza dell'Informazione
Via Sommarive 9, I-38123 Povo (TN), Italy

Email: michela.ceria@unitn.it

Manuscript received 20 March 2015; communicated 15 June 2015; published online 11 October 2016