NB.06.F3

July 23, 2024

1 Proposition

Let t be a line of $\mathbb{P}^2$. * If t is not tangent to the isotropic conic, $t \subseteq \mathrm{Eig}(C)$ for a cubic C if and only $C = t^2\ell$, where ℓ is any line of the plane. * If t is tangent to the isotropic conic (in a point~P), $t \subseteq \mathrm{Eig}(C)$ for a cubic C if and only if

$$C = t^2\ell + \lambda C(r_0),$$

where ℓ is any line of the plane, $\lambda \in \mathbb{C}$, r_0 is any fixed line passing through P different from t and $C(r_0)$ is defined by

$$C(r) = \left(r^2 - 3\left(a^2 + b^2 + c^2\right) \mathcal{Q}\mathrm{iso}\right) r$$

Moreover, if t is tangent to the isotropic conic at P, any cubic C with $t \subseteq \mathrm{Eig}(C)$ is singular in P and t is one of the tangents C in P.

[1]:
```
load("basic_functions.sage")
```

1.1 If a cubic C has an eigenline t tangent to the isotropic conic, then the equation of C is F_1, so we have a linear system of cubics of dimension 3.

1.2 This linear system is of the form $t^2\ell + \lambda C(r)$ for any fixed line r different from t. Hence the family of cubics F_1 and the family of cubics $t^2\ell + \lambda C(r)$ coincide.

We define the point P_1 and the line tg1, tangent to Ciso in P_1, the eigenpoint locus of the generic cubic F of the plane

[2]:
```
P1 = vector((1, ii, 0))
tg1 = x+ii*y
eigF = eig(F)
```

In order to have that the points of tg1 are eigenpoints, we substitute $x = -iy$ into eigF and we get a list of three equations that have to be satisfied for all values of x

[3]:
```
EE = list(eigF.subs(x = -ii*y))
```

We construct the list eqs of equations in $a_0, \dots, a_9$ that must be satisfied in order to have tg1 eigenline:

[4]:
```
eqs = [ee.coefficient(mm) for mm in mon for ee in EE]
```

In order to obtain the cubics F_1 which have tg1 as eigenline, we reduce F w.r.t. the ideal (eqs).

[5]:
```
F1 = S.ideal(eqs).reduce(F)
```

F_1 has the variables $x, y, z, a_2, a_3, a_6, a_9$, hence is a linear space of dimension 3 in $\mathbb{P}^9$ generated by four cubics which are the polynomials G_1, G_2, G_3, G_4 below:

[6]:
```
assert(F1.variables() == (x, y, z, a2, a3, a6, a9))

G1, G2, G3, G4 = (
    F1.coefficient(a2), F1.coefficient(a3),
    F1.coefficient(a6), F1.coefficient(a9)
)
```

We define four other cubics which are H_1, H_2, H_3 as follows:

[7]:
```
H1, H2, H3 = x*tg1^2, y*tg1^2, z*tg1^2
```

and H_4, constructed in this way: we take the generic line r passing through the point P_1 , (which is $l_1 tg1 + l_2 z$) and H_4 is the corresponding cubic whose eigenpoints are the two tangent lines to Ciso in the points $r \cap Ciso$, which is given by $C(r)$, i.e. the formula $(r^2 - 3(a^2 + b^2 + c^2)Ciso)r$ (where $a = l_1, b = il_1, c = il_2$)

[8]:
```
r = l1*tg1+l2*z
H4 = r*(r^2-3*(r.coefficient(x)^2+r.coefficient(y)^2+r.coefficient(z)^2)*Ciso)
```

We verify that H_1, H_2, H_3, H_4 are 4 linearly independent cubics, i.e. we see that the rank of the matrix N_1 below is always 4 (unless $l_2 = 0$, i.e. unless tg1 = r)

[9]:
```
N1 = matrix([[hh.coefficient(mn) for mn in mon] for hh in [H1, H2, H3, H4]])
JN1 = S.ideal(N1.minors(4))
assert(JN1.radical() == S.ideal(l2)) ## N1 has always rank 4
```

We verify that the linear space generated by G_1, G_2, G_3, G_4 coincides with the linear space generated by H_1, H_2, H_3, H_4

[10]:
```
for hh in [H1, H2, H3, H4]:
    assert(
        matrix(
            [
                [gg.coefficient(mn) for mn in mon] for gg in [G1, G2, G3, G4] +␣
 ↪[hh.coefficient(mn) for mn in mon]
            ]
        ).rank() == 4
    )

for gg in [G1, G2, G3, G4]:
    assert(
        matrix(
            [
                [hh.coefficient(mn) for mn in mon] for hh in [H1, H2, H3, H4] +␣
 ↪[gg.coefficient(mn) for mn in mon]
```

```
            ]
        ).rank() == 4
    )
```

1.3 All the cubics F_1 are singular in P_1 and that tg1 is tangent to F_1 in P_1.

1.4 We assume a_9 not zero, since if $a_9 = 0$, F_1 is reducible of the form $t^2\ell$, where t is the line tg1

[11]:
```
assert(F1.subs(a9=0).quo_rem((x+ii*y)^2)[1] == 0)  ## (x+ii*y)^2 is a factor of
 ↪the cubic
assert(gdn(F1).subs(substitution(P1)) == vector(S, (0,0,0)))  ## P1 is singular
assert(S.ideal(F1, tg1).saturation(a9)[0] == S.ideal(tg1, z^3))  ## P1 is a
 ↪triple point
```

In order to describe F_1, we first determine when F_1 splits into a line and a conic. The line necessarily passes through P_1, hence is the line r above. A generic point of r is $(u_1 l_2, u_2 l_2, -l_1(u_1 + iu_2))$, hence we define the point p_3:

[12]:
```
p3 = vector(S, (u1*l2, u2*l2, -l1*(u1+ii*u2)))
assert(r.subs(substitution(p3)) == 0)
```

Now we impose that, for all u_1 and u_2, the point p_3 is a zero of F_1. With this substitution F_1 splits into $(u_1 + iu_2)^2$ and another factor, linear in u_1 and u_2 and we want that for all u_1 and u_2 this factor is zero:

[13]:
```
ff1 = F1.subs(substitution(p3))
assert(ff1.quo_rem((u1+ii*u2)^2)[1] == 0)
ff2 = ff1.quo_rem((u1+ii*u2)^2)[0]
assert(ff2.degree(u1) ==1)
assert(ff2.degree(u2) == 1)
jf1 = S.ideal(ff2.coefficient(u1), ff2.coefficient(u2))
```

The ideal jf1 is the ideal of the conditions on the parameters of F_1 (which are a_2, a_3, a_6, a_9) in order to have that $V(F_1)$ splits into the line r and a conic. We can saturate it w.r.t. a_9 and l_2. We get an ideal jf2 which is generated by the three polynomials h_1, h_2, h_3 below:

[14]:
```
jf2 = jf1.saturation(a9*l2)[0]
h1 = l2*a2 + (3*ii)*l2*a3 + 3*l1*a9
h2 = l1*a2^2 + (6*ii)*l1*a2*a3 - 9*l1*a3^2 + (-9*ii)*l2*a3*a9 - 9*l1*a6*a9
h3 = a2^3 + (9*ii)*a2^2*a3 - 27*a2*a3^2 + (-27*ii)*a3^3 - 9*a2*a6*a9 +
 ↪(-27*ii)*a3*a6*a9 + (27*ii)*a3*a9^2
assert(S.ideal(h1, h2, h3).saturation(a9)[0] == jf2)
```

The meaning of the above computation is the following: $V(F_1)$ contains a line r iff a_2, a_3, a_6, a_9 satisfy the condition $h_3 = 0$. (The values of l_1 and l_2 which determine the line r can be computed from the linear system obtained from h_1 and h_2). Hence:

F_1 splits into the product of a linear factor and a quadratic factor iff $h_3(a_2, a_3, a_6, a_9) = 0$.

Now a similar computation: we want to see when a line passing thorugh P_1 is tangent to the cubic $V(F_1)$ in P_1. Again, the line is the line r above, we consider the ideal generated by F_1 and r and we study it. We compute its primary decomposition and we get two ideals: the radical of the first is the point P_1, the second is generated by two poynomials g_1 and g_2 linear in x, y, z (up to an admissible saturation) (g_1 is r) and g_1, g_2 give therefore a further point of intersectrion of r with the cubic $V(F_1)$:

[15]:
```
J1 = S.ideal(F1, r)
pd = J1.primary_decomposition()
assert(len(pd) == 2)
assert(pd[0].radical() == S.ideal(x+ii*y, z))
g1 = pd[1].gens()[0]
g2 = pd[1].gens()[1]
assert(g1 == r)
assert(g1.degree(x) == 1)
assert(g1.degree(y) == 1)
assert(g1.degree(z) == 1)
assert(g2.degree(x) == 1)
assert(g2.degree(y) == 1)
assert(g2.degree(z) == 1)
assert(pd[1] == S.ideal(g1, g2).saturation(l2)[0])
```

At this point we know that the third intersection of the line r with $V(F_1)$ is given by the linear system $g_1 = 0, g_2 = 0$. We want that this point (for suitable values of l_1 and l_2) is again the point P_1. In this way we determine when r is tangent to $V(F_1)$. Since g_1 is r, it always contain P_1. The only condition that has to be verified is that P_1 annihilates g_2. This condition is $f = l_2a_2 + 3il_2a_3 + 3l_1a_9 = 0$:

[16]:
```
assert(g1.subs(substitution(P1)) == 0)
f = l2*a2 + (3*ii)*l2*a3 + 3*l1*a9
assert(g2.subs(substitution(P1)) == l2 * f)
```

Hence we obtain that r is tangent in P_1 to $V(F_1)$ when l_1, l_2 are such that $f = 0$. The tangent is r_2 (we do not get tg1, because we exclude the case $l_2 = 0$):

[17]:
```
r2 = r.subs({l1:f.coefficient(l2), l2: -f.coefficient(l1)})
```

In general, the line r_2 is the second tangent in P_1 to the cubic. To verify this, we study the ideal generated by F_1 and r_2. Its primary decomposition (after saturation w.r.t. a_9) has two components, the first is the triple point P_1, the second contains the polynomial h_3, which means that it appears only when F_1 is reducible.

[18]:
```
pd2 = S.ideal(F1, r2).saturation(a9)[0].primary_decomposition()
assert(len(pd2) == 2)
assert(z^3 in pd2[0])
assert(pd2[0].radical() == S.ideal(x+ii*y, z))  ## pd2[0] is the triple point
 ↪$P_1$.
assert(h3 in pd2[1])
```

Hence, when F_1 is irreducible, $V(F_1)$ has two tangents in P_1, which are: tg1 and r2. We want to see when they are coincident, i.e. when P_1 is a cusp. Here we see that this happens iff $a_9 = 0$, i.e. iff the cubic is splits into $t^2\ell$, where t is the line tg1. Therefore:

If $V(F_1)$ is irreducible, it has two distinct tangents tg1 and r_2, so the point P_1 is never a cusp for the cubic.

[19]:
```
coinc_l = S.ideal(matrix([[tg1.coefficient(vr) for vr in [x, y, z]], [r2.
  ↪coefficient(vr) for vr in [x, y, z]]]).minors(2))
assert(coinc_l.groebner_basis() == [a9])
assert(F1.subs(a9=0).quo_rem((x+ii*y)^2)[1] == 0)
```

When $V(F_1)$ is reducible, i.e. when $h_3 = 0$, F_1 splits into the product of r_2 and a conic. We can easily verify this, since h_3 is linear in a_6, so the equation $h_3 = 0$ can be solved w.r.t. a_6

[20]:
```
assert(h3.degree(a6) == 1)
st = {a6: -h3.subs(a6=0)/h3.coefficient(a6)}
F2 = S(numerator(F1.subs(st)))
assert(F2.quo_rem(r2)[1] == 0)
```