NB.04.F6

July 23, 2024

1 Proposition

If the five points $P_1, \ldots, P_5$ satisfy

$$\delta_1(P_1, P_2, P_4) = \overline{\delta}_1(P_1, P_2, P_3) = \overline{\delta}_1(P_1, P_4, P_5) = 0$$

and if we impose the condition that there is an eigenpoint, say P_6, aligned with P_2 and P_4, then the eigenpoints satisfy all these alignments:

$$(P_1, P_2, P_3), (P_1, P_4, P_5), (P_2, P_4, P_6), (P_2, P_5, P_7), (P_3, P_4, P_7), (P_3, P_5, P_6)$$

Hence the points P_6 and P_7 are determined by $P_1, \ldots, P_5$ since $P_6 = (P_2 \vee P_4) \cap (P_3 \vee P_5)$ and $P_7 = (P_3 \vee P_4) \cap (P_2 \vee P_5)$. A similar result holds if we take P_3 in place of P_2 or P_5 in place of P_4.

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

2 $\delta_1(P_1, P_2, P_4) = 0$.

P_1 cannot be on the isotropic conic, so we can assume $P_1 = (1 : 0 : 0)$.

[2]:
```
P1 = vector((1, 0, 0))
P2 = vector(S, (A2, B2, C2))
P4 = vector(S, (A4, B4, C4))
```

We define P_3 and P_5 according to the Propositioin 4.8

[3]:
```
P3 = (
    (scalar_product(P1, P2)^2+scalar_product(P1, P1)*scalar_product(P2, P2))*P1
    - 2*(scalar_product(P1, P1)*scalar_product(P1, P2))*P2
)
P5 = (
    (scalar_product(P1, P4)^2+scalar_product(P1, P1)*scalar_product(P4, P4))*P1
    - 2*(scalar_product(P1, P1)*scalar_product(P1, P4))*P4
)
```

We have: $\overline{\delta}_1(P_1, P_2, P_3) = 0$, $\overline{\delta}_1(P_1, P_4, P_5) = 0$, $\delta_2(P_1, P_2, P_3, P_4, P_5) = 0$

[4]:
```
assert(delta1b(P1, P2, P3) == 0)
assert(delta1b(P1, P4, P5) == 0)
assert(delta2(P1, P2, P3, P4, P5) == 0)
```

Remember that we also must have $\delta_1(P_1, P_2, P_4) = 0$, which means that $B_2B_4 + C_2C_4 = 0$

[5]:
```
D = B2*B4 + C2*C4
assert(delta1(P1, P2, P4) == B2*B4 + C2*C4)
```

We define a generic point PP on the line $P_2 \vee P_4$

[6]:
```
PP = w1*P2+w2*P4
```

We know that the condition matrix of P_1, P_2, P_3, P_4, P_5 is of rank 8. We select a submatrix of it of rank 8

[7]:
```
M = condition_matrix([P1, P2, P3, P4, P5], S, standard="all")
```

[8]:
```
M1 = M.matrix_from_rows([0, 1, 3, 4, 6, 7, 9, 10])
assert(M1.rank() == 8)
```

If PP is an eigenpoint, the determinant of the three matrices below must be zero:

[9]:
```
Ma = M1.stack(phi(PP, S)[0]).stack(phi(PP, S)[1])
Mb = M1.stack(phi(PP, S)[0]).stack(phi(PP, S)[2])
Mc = M1.stack(phi(PP, S)[1]).stack(phi(PP, S)[2])
```

[10]:
```
da = Ma.det()
db = Mb.det()
dc = Mc.det()
```

Now we consider the ideal of da, db, dc plus the condition $\delta_1(P_1, P_2, P_4) = 0$ and we saturate it as much as possible

[11]:
```
J = S.ideal(da, db, dc, delta1(P1, P2, P4)).saturation(w1*w2)[0]
J = J.saturation(matrix([P1, P2, P4]).det())[0]
J = J.saturation(S.ideal(matrix([P1, P2]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([P1, P3]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([P2, P3]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([P1, P4]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([P1, P5]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([P2, P5]).minors(2)))[0]
```

The ideal J is prime and is generated by: $\delta_1(P_1, P_2, P_4), F$, where

$$F = 2w_1A_2^2A_4 + w_1B_2^2A_4 + w_1C_2^2A_4 + 2w_2A_2A_4^2 + w_2A_2B_4^2 + w_2A_2C_4^2$$

[12]:
```
assert(J.is_prime())
F = 2*w1*A2^2*A4 + w1*B2^2*A4 + w1*C2^2*A4 + 2*w2*A2*A4^2 + w2*A2*B4^2 +
 ↪w2*A2*C4^2
assert(J == S.ideal(F, delta1(P1, P2, P4)))
```

F is a polynomial linear in w_1, w_2 and the values of w_1 and w_2 such that $F = 0$ give a point PP which is candiate to be an eigenpoint.

Remember that $\delta_1(P_1, P_2, P_4) = B_2B_4 + C_2C_4$ must be zero, so we have two possibilities:

2.1 Case 1:

$B_2 = 0$ and $C_4 = 0$

[13]:
```
sst = {B2:0, C4:0}
p1 = P1.subs(sst)
p2 = P2.subs(sst)
p3 = P3.subs(sst)
p4 = P4.subs(sst)
p5 = P5.subs(sst)
pp = PP.subs(sst)

F1 = F.subs(sst)
```

[14]:
```
assert(F1.subs({w1: F1.coefficient(w2), w2: -F1.coefficient(w1)}) == 0)
p6 = pp.subs({w1: F1.coefficient(w2), w2: -F1.coefficient(w1)})
```

The following computation shows that p_6 is an eigenpoint:

[15]:
```
mm = condition_matrix([p1, p2, p3, p4, p5, p6], S, standard="all")
assert(mm.rank() == 9)
```

The point p_6 is always define, i.e. it is not possible to have that the coefficients of w_1 and of w_2 in F_1 are both zero.

[16]:
```
J = S.ideal(F1.coefficient(w1), F1.coefficient(w2))
J = J.saturation(S.ideal(matrix([p1, p2]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([p1, p4]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([p1, p5]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([p2, p3]).minors(2)))[0]
assert(J == S.ideal(1))
```

We define a point p_7 as the intersection point of the lines $p_3 \vee p_4$ and $p_2 \vee p_5$ and we verify that it is an eigenpoint

[17]:
```
p7 = vector(S, intersection_lines(p3, p4, p2, p5))
mm = condition_matrix([p1, p2, p3, p4, p5, p7], S, standard="all")
assert(mm.rank() == 9)
```

Now we see that the points $p_1, \ldots, p_7$ have the alignments:

$$[(1, 2, 3), (1, 4, 5), (2, 4, 6), (2, 5, 7), (3, 4, 7), (3, 5, 6)]$$

[18]:
```
assert(
    alignments([p1, p2, p3, p4, p5, p6, p7])
    == [(1, 2, 3), (1, 4, 5), (2, 4, 6), (2, 5, 7), (3, 4, 7), (3, 5, 6)]
)
```

Hence the points are in configuration (8). Note that p_1, p_6, p_7 are not aligned, although we have $\delta_2(p_1, p_2, p_3, p_4, p_5) = 0$.

We have the following orthogonalities: * $p_1 \vee p_4$ orthogonal to $p_1 \vee p_2$, * $p_2 \vee p_4$ orthogonal to $p_3 \vee p_5$, * $p_2 \vee p_5$ orthogonal to $p_3 \vee p_4$.

[19]:
```
assert(scalar_product(wedge_product(p1, p4), wedge_product(p1, p2))==0)
assert(scalar_product(wedge_product(p2, p4), wedge_product(p3, p5))==0)
assert(scalar_product(wedge_product(p2, p5), wedge_product(p3, p4))==0)
```

Here we verify that p_4 can be obtained from p_2, p_5, p_7 via the formula:

$$p_4 = (p_2 \times p_3)s_{25}s_{35} - s_{23}(p_2 \times p_5)s_{35} + s_{23}s_{25}(p_3 \times p_5)$$

[20]:
```
q4 = (
    wedge_product(p2, p3)*scalar_product(p2, p5)*scalar_product(p3, p5)
    - scalar_product(p2, p3)*wedge_product(p2, p5)*scalar_product(p3, p5)
    + scalar_product(p2, p3)*scalar_product(p2, p5)*wedge_product(p3, p5)
)
```

[21]:
```
assert(matrix([p4, q4]).minors(2) == [0, 0, 0])
```

2.2 Case 2:

$B_4 = -C_2C_4/B_2$

[22]:
```
sst = {B4: -C2*C4, C4: C4*B2}
```

[23]:
```
assert(D.subs(sst) == 0)
```

[24]:
```
p1 = P1.subs(sst)
p2 = P2.subs(sst)
p3 = P3.subs(sst)
p4 = P4.subs(sst)
p5 = P5.subs(sst)
pp = PP.subs(sst)

F2 = F.subs(sst)
```

[25]:
```
assert(F2.subs({w1: F2.coefficient(w2), w2: -F2.coefficient(w1)}) == 0)
p6 = pp.subs({w1: F2.coefficient(w2), w2: -F2.coefficient(w1)})
```

The following computation shows that p_6 is an eigenpoint:

[26]:
```
mm = condition_matrix([p1, p2, p3, p4, p5, p6], S, standard="all")
assert(mm.rank() == 9)
```

The point p_6 is always defined, i.e. it is not possible to have that the coefficients of w_1 and of w_2 in F_1 are both zero.

[27]:
```
J = S.ideal(F2.coefficient(w1), F2.coefficient(w2))
J = J.saturation(S.ideal(matrix([p1, p2]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([p1, p4]).minors(2)))[0]
```

```
J = J.saturation(S.ideal(matrix([p1, p5]).minors(2)))[0]
J = J.saturation(S.ideal(matrix([p2, p3]).minors(2)))[0]
assert(J == S.ideal(1))
```

We define a point p_7 as the intersection point of the lines $p_3 \vee p_4$ and $p_2 \vee p_5$ and we verify that it is an eigenpoint

[28]:
```
p7 = vector(S, intersection_lines(p3, p4, p2, p5))
mm = condition_matrix([p1, p2, p3, p4, p5, p7], S, standard="all")
assert(mm.rank() == 9)
```

Now we see that the points $p_1, \ldots, p_7$ have the alignments:

$$[(1,2,3),(1,4,5),(2,4,6),(2,5,7),(3,4,7),(3,5,6)]$$

[29]:
```
assert(
    alignments([p1, p2, p3, p4, p5, p6, p7])
    == [(1, 2, 3), (1, 4, 5), (2, 4, 6), (2, 5, 7), (3, 4, 7), (3, 5, 6)]
)
```

Hence the points are in configuration (8). Note that p_1, p_6, p_7 are not aligned, although we have $\delta_2(p_1, p_2, p_3, p_4, p_5) = 0$.

we have the following orthogonalities: $p_1 \vee p_4$ ort to $p_1 \vee p_2$, $p_2 \vee p_4$ ort to $p_3 \vee p_5$, $p_2 \vee p_5$ ort to $p_3 \vee p_4$

[30]:
```
assert(scalar_product(wedge_product(p1, p4), wedge_product(p1, p2))==0)
assert(scalar_product(wedge_product(p2, p4), wedge_product(p3, p5))==0)
assert(scalar_product(wedge_product(p2, p5), wedge_product(p3, p4))==0)
```

Here we verify that p_4 can be obtained from p_2, p_5, p_7 via the formula:

$$p_4 = (p_2 \times p_3)s_{25}s_{35} - s_{23}(p_2 \times p_5)s_{35} + s_{23}s_{25}(p_3 \times p_5)$$

[31]:
```
q4 = (
    wedge_product(p2, p3)*scalar_product(p2, p5)*scalar_product(p3, p5)
    - scalar_product(p2, p3)*wedge_product(p2, p5)*scalar_product(p3, p5)
    + scalar_product(p2, p3)*scalar_product(p2, p5)*wedge_product(p3, p5)
)
```

[32]:
```
assert(matrix([p4, q4]).minors(2) == [0, 0, 0])
```