# Section 3

In this notebook we explain Example 3.2 and how we obtain the list of polygons.
Points are as in Example 3.2. We work in the real affine 3-space, and equivalently, the real projective plane.

In[152]:=

$$\text{pts} = \text{Transpose}\left[\begin{pmatrix} -2 & 24 & 16 & 27 & 14 & 1 \\ -25 & 3 & 28 & 13 & 5 & 7 \\ -26 & -4 & 1 & -14 & 9 & 6 \end{pmatrix}\right];$$

We generate homogeneous equations for (projective) lines and conics through these points.

In[270]:=

```
lines = Map[Det[{#[[1]], #[[2]], {x, y, z}}] &, Subsets[pts, {2}]];
conics = Map[Det[Map[DeleteDuplicates[Flatten[Outer[Times, #, #]]] &,
      Catenate[{#, {{x, y, z}}}]]] &, Subsets[pts, {5}]];
fns =
  Map[Simplify[# / Apply[GCD, Flatten[CoefficientList[#, {x, y, z}]]]] &,
   Catenate[{lines, conics}]];
```

In[156]:=

```
fnsName =
  {"F12", "F13", "F14", "F15", "F16", "F23", "F24", "F25", "F26", "F34",
   "F35", "F36", "F45", "F46", "F56", "G6", "G5", "G4", "G3", "G2", "G1"};
ptsName = {"E1", "E2", "E3", "E4", "E5", "E6"};
```

We produce realizable sign patterns with respect to these expressions using cylindrical decomposition. They will give us the regions in the real affine 3-space that are cut out by these equations.

In[364]:=
```
RealizableSigns[fns_, vars_] :=
  Reap[RealizableSigns[fns, vars, {}, True]][[2, 1]];
RealizableSigns[fns_, vars_, sig_, cad_] := Module[{fn, cadp, cadm},
   If[Length[sig] == Length[fns],
    Sow[sig];
    Return[]
   ];
   fn = fns[[Length[sig] + 1]];

   cadp = CylindricalDecomposition[cad && fn > 0, vars];
   If[FindInstance[cadp, vars, Reals] ≠ {},
    RealizableSigns[fns, vars, Catenate[{sig, {+1}}], cadp]];

   cadm = CylindricalDecomposition[cad && fn < 0, vars];
   If[FindInstance[cadm, vars, Reals] ≠ {},
    RealizableSigns[fns, vars, Catenate[{sig, {-1}}], cadm]];
  ];

signs = RealizableSigns[fns, {x, y, z}];
```

Check that we have 260 realizable sign vectors. This is expected because projectively they collapse to 130 equivalence classes, which correspond to the 130 regions in the real projective plane.

In[370]:=
```
Length[signs]
```

Out[370]=
```
260
```

Two sign vectors are equivalent projectively if and only if they differ in the first 15 coordinates, which correspond to the linear forms, and agree in the last 6 coordinates, which correspond to the degree two forms.

In[372]:=
```
opposite = {-1, -1, -1, -1, -1, -1, -1,
   -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1};
dualSign[sign_] := Inner[#1 * #2 &, sign, opposite, List];
regions =
  Table[Select[Table[{signs[[i]], dualSign[signs[[i]]]}, {i, 1, 260}],
     LexicographicOrder[#[[1]], #[[2]]] === 1 &][[j]], {j, 1, 130}];
```

We obtain the boundary of a region in the projective plane by first finding its neighbours, i.e., the regions whose sign vectors differ in exactly one entry, and then recording the index of the entries in which they differ.

In[378]:=
```
neighbours[sign_] := Reap[For[i = 1, i < 261, i++,
      If[Count[Inner[#1 + #2 &, signs[[i]], sign, List], 0] === 1,
       Sow[i]
      ]
     ]
    ][[2]][[1]];
boundary[sign_] :=
  Position[Select[Table[Inner[#1 + #2 &, signs[[i]], sign, List],
      {i, 1, Length[signs]}], Count[##, 0] === 1 &], 0][[All, 2]];
```

For example, region 7 has four neighbors and each of them differ in entry 9, 20, 13, 6, respectively.

In[381]:=
```
neighbours[regions[[7]][[1]]]
boundary[regions[[7]][[1]]]
```

Out[381]= {131, 138, 139, 141}

Out[382]= {9, 20, 13, 6}

Each region has two representative sign vectors. We verify that the output is consistent using the other representative.

In[377]:=
```
boundary[regions[[7]][[2]]]
```

Out[377]= {6, 13, 20, 9}

The “lines” on the boundary of the region in the projective plane are given by the following.

In[383]:=
```
fnsName[[boundary[regions[[7]][[1]]]]]
```

Out[383]= {F26, G2, F45, F23}

Regions on the cubic surface are in bijection with the regions on the plane. However, on the cubic surface, there are six more lines that come from the blow up of the six points. In particular, a polygon on the surface will have one more side on the boundary for each exceptional point that lies on the vertex of the polygon in the plane.

Therefore, we must find out which polygons in the plane have exceptional points as vertices. We record the sign vectors for the points.

```
exceptional =
  Table[Map[Sign[#] &, fns /. Thread[{x, y, z} → pts⟦i⟧]], {i, 1, 6}]
```

Out[216]=
```
{{0, 0, 0, 0, 0, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 0, 0, 0, 0, 0, 1},
 {0, 1, 1, -1, 1, 0, 0, 0, 0, -1, 1, 1, 1, 1, -1, 0, 0, 0, 0, -1, 0},
 {-1, 0, -1, -1, 1, 0, 1, -1, -1, 0, 0, 0, -1, -1, -1, 0, 0, 0, 1, 0, 0},
 {-1, 1, 0, -1, 1, -1, 0, -1, -1, 0, 1, 1, 0, 0, -1, 0, 0, -1, 0, 0, 0},
 {1, 1, 1, 0, 1, 1, 1, 0, 1, -1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0},
 {-1, -1, -1, -1, 0, 1, 1, -1, 0, -1, -1, 0, -1, 0, 0, -1, 0, 0, 0, 0, 0}}
```

Let $v = (v_i)$ and $w = (w_i)$ be sign vectors where 0 is allowed. We say v and w are weakly equal, denoted $v \approx w$, if $v_i = w_i$ or $0 \in \{v_i, w_i\}$ for all i. For an exceptional point to be a vertex of a polygon, a necessary (but not sufficient) condition is that it satisfies the equation of two sides of the polygon and its sign vector weakly equals the sign vector of the region. We also check the projectively equivalent sign vector for the exceptional points.

In[384]:=
```
checkPair[sign_, indices_] := Reap[Module[{i, candidates, j},
    candidates = Flatten[
      Position[Table[exceptional⟦i⟧⟦indices⟧, {i, 1, 6}], {0, 0}]];
    For[j = 1, j < Length[candidates] + 1, j++,
     If[
      Count[Inner[#1 + #2 &, exceptional⟦candidates⟦j⟧⟧, sign, List], 0] ===
        0 || Count[Inner[#1 + #2 &, dualSign[exceptional⟦candidates⟦j⟧⟧],
          sign, List], 0] === 0 ,
      Sow[candidates⟦j⟧]
     ]
    ]
   ]]⟦2⟧
```

For example, we check whether any exceptional point lies on the boundary of region 7. And we find none.

In[385]:=
```
Map[checkPair[regions⟦7⟧⟦1⟧, #] &, Subsets[boundary[regions⟦7⟧⟦1⟧], {2}]]
```

Out[385]=
```
{{}, {}, {}, {}, {}, {}}
```

We now produce a full list of polygons. However, note that the condition we mentioned above is only necessary but not sufficient. It turns out that in six polygons, an exceptional point is falsely added to their boundaries.

In[386]:=
```
allPolygons = Table[Sort[Join[fnsName⟦boundary[allRegions⟦i⟧⟦1⟧]⟧, ptsName⟦
      DeleteDuplicates[Flatten[Map[checkPair[allRegions⟦i⟧⟦1⟧, #] &,
         Subsets[boundary[allRegions⟦i⟧⟦1⟧], {2}]]]]⟧]], {i, 1, 130}];
```

In[387]:=
```
Sort[allPolygons]
```

Out[387]=
```
{{E4, F24, G2}, {E4, F34, G3}, {E5, F15, G1}, {E5, F25, G2},
 {E6, F16, G1}, {E6, F36, G3}, {F14, F25, F36}, {F14, F26, F35},
 {F15, F26, F34}, {F16, F24, F35}, {E1, E5, G3, G4}, {E1, E5, G3, G6},
 {E1, E6, F16, G5}, {E1, E6, G2, G4}, {E1, E6, G2, G5}, {E1, E6, G3, G4},
 {E1, F12, F14, F36}, {E1, F12, F14, G1}, {E1, F12, F15, F36},
 {E1, F13, F14, G1}, {E1, F13, F16, F25}, {E1, F15, F36, G6},
 {E1, F16, F25, G5}, {E2, E4, G1, G5}, {E2, E4, G1, G6}, {E2, E4, G3, G6},
 {E2, E5, F25, G4}, {E2, E5, G1, G6}, {E2, E5, G3, G4}, {E2, E5, G3, G6},
 {E2, F12, F25, F34}, {E2, F12, F26, F34}, {E2, F12, F26, G2},
 {E2, F15, F23, F24}, {E2, F15, F24, G5}, {E2, F23, F26, G2},
 {E2, F25, F34, G4}, {E3, E4, G1, G5}, {E3, E4, G1, G6}, {E3, E4, G2, G5},
 {E3, E6, G2, G4}, {E3, E6, G2, G5}, {E3, F13, F24, F36}, {E3, F13, F35, G3},
 {E3, F16, F23, F34}, {E3, F16, F23, F35}, {E3, F16, F34, G6},
 {E3, F23, F35, G3}, {E3, F24, F36, G4}, {E4, F14, F23, F45},
 {E4, F14, F23, F46}, {E4, F14, F45, G4}, {E4, F14, F46, G4},
 {E4, F24, F45, G4}, {E4, F34, F46, G4}, {E5, F12, F35, F45},
 {E5, F12, F35, F56}, {E5, F12, F45, G2}, {E5, F15, F56, G5},
 {E5, F35, F45, G5}, {E5, F35, F56, G5}, {E6, F13, F26, F46},
 {E6, F13, F26, F56}, {E6, F13, F56, G1}, {E6, F26, F46, G6},
 {E6, F26, F56, G6}, {E6, F36, F46, G6}, {F12, F14, F56, G1},
 {F12, F15, F36, F46}, {F12, F25, F34, F46}, {F12, F26, F35, F45},
 {F12, F26, F45, G2}, {F13, F14, F26, F56}, {F13, F14, F56, G1},
 {F13, F16, F24, F45}, {F13, F16, F25, F45}, {F13, F24, F36, F45},
 {F13, F25, F36, F45}, {F13, F35, F46, G3}, {F14, F23, F35, F46},
 {F14, F25, F46, G4}, {F15, F23, F24, F56}, {F15, F23, F34, F56},
 {F15, F24, F56, G5}, {F15, F26, F46, G6}, {F15, F36, F46, G6},
 {F16, F23, F34, F56}, {F16, F25, F45, G5}, {F16, F34, F56, G6},
 {F23, F26, F45, G2}, {F23, F35, F46, G3}, {F24, F35, F56, G5},
 {F24, F36, F45, G4}, {F25, F34, F46, G4}, {E1, E5, F12, F15, G1},
 {E1, E5, F15, G1, G6}, {E1, E5, F25, G2, G4}, {E1, E5, F25, G2, G5},
 {E1, E6, F13, F16, G1}, {E1, E6, F36, G3, G6}, {E1, F13, F14, F25, F36},
 {E2, E4, F23, F24, G2}, {E2, E4, F24, G2, G5}, {E2, E4, F34, G3, G4},
 {E2, E5, F12, F25, G2}, {E2, E5, F15, G1, G5}, {E2, F15, F23, F26, F34},
 {E3, E4, F23, F34, G3}, {E3, E4, F24, G2, G4}, {E3, E4, F34, G3, G6},
 {E3, E6, F13, F36, G3}, {E3, E6, F16, G1, G5}, {E3, E6, F16, G1, G6},
 {E3, E6, F36, G3, G4}, {E3, F13, F16, F24, F35}, {E4, F23, F24, F45, G2},
 {E4, F23, F34, F46, G3}, {E5, F12, F15, F56, G1}, {E5, F25, F45, G2, G5},
 {E6, F13, F36, F46, G3}, {E6, F16, F56, G1, G6}, {F12, F14, F25, F36, F46},
 {F12, F14, F26, F35, F56}, {F12, F15, F26, F34, F46},
 {F13, F14, F26, F35, F46}, {F14, F23, F26, F35, F45},
 {F14, F25, F36, F45, G4}, {F15, F26, F34, F56, G6},
 {F16, F23, F24, F35, F56}, {F16, F24, F35, F45, G5}}
```

There are six pentagons that are really quadrilateral. They are
{“E1”, “E5”, “F25”, “G2”, “G5”},

{“E1”, “E6”, “F36”, “G3”, “G6”},
{“E2”, “E4”, “F34”, “G3”, “G4”},
{“E2”, “E5”, “F15”, “G1”, “G5”},
{“E3”, “E4”, “F24”, “G2”, “G4”},
and {“E3”, “E6”, “F16”, “G1”, “G6”}.
In each of these cases, the exceptional divisor that meets the unique line $F_{ij}$ does not lie on the boundary of the region.

In[316]:=
```
truePolygons =
  Sort[{{"E4", "F24", "G2"}, {"E4", "F34", "G3"}, {"E5", "F15", "G1"},
    {"E5", "F25", "G2"}, {"E6", "F16", "G1"}, {"E6", "F36", "G3"},
    {"F14", "F25", "F36"}, {"F14", "F26", "F35"}, {"F15", "F26", "F34"},
    {"F16", "F24", "F35"}, {"E1", "E5", "G3", "G4"},
    {"E1", "E5", "G3", "G6"}, {"E1", "E6", "F16", "G5"},
    {"E1", "E6", "G2", "G4"}, {"E1", "E6", "G2", "G5"},
    {"E1", "E6", "G3", "G4"}, {"E1", "F12", "F14", "F36"},
    {"E1", "F12", "F14", "G1"}, {"E1", "F12", "F15", "F36"},
    {"E1", "F13", "F14", "G1"}, {"E1", "F13", "F16", "F25"},
    {"E1", "F15", "F36", "G6"}, {"E1", "F16", "F25", "G5"},
    {"E2", "E4", "G1", "G5"}, {"E2", "E4", "G1", "G6"},
    {"E2", "E4", "G3", "G6"}, {"E2", "E5", "F25", "G4"},
    {"E2", "E5", "G1", "G6"}, {"E2", "E5", "G3", "G4"},
    {"E2", "E5", "G3", "G6"}, {"E2", "F12", "F25", "F34"},
    {"E2", "F12", "F26", "F34"}, {"E2", "F12", "F26", "G2"},
    {"E2", "F15", "F23", "F24"}, {"E2", "F15", "F24", "G5"},
    {"E2", "F23", "F26", "G2"}, {"E2", "F25", "F34", "G4"},
    {"E3", "E4", "G1", "G5"}, {"E3", "E4", "G1", "G6"},
    {"E3", "E4", "G2", "G5"}, {"E3", "E6", "G2", "G4"},
    {"E3", "E6", "G2", "G5"}, {"E3", "F13", "F24", "F36"},
    {"E3", "F13", "F35", "G3"}, {"E3", "F16", "F23", "F34"},
    {"E3", "F16", "F23", "F35"}, {"E3", "F16", "F34", "G6"},
    {"E3", "F23", "F35", "G3"}, {"E3", "F24", "F36", "G4"},
    {"E4", "F14", "F23", "F45"}, {"E4", "F14", "F23", "F46"},
    {"E4", "F14", "F45", "G4"}, {"E4", "F14", "F46", "G4"},
    {"E4", "F24", "F45", "G4"}, {"E4", "F34", "F46", "G4"},
    {"E5", "F12", "F35", "F45"}, {"E5", "F12", "F35", "F56"},
    {"E5", "F12", "F45", "G2"}, {"E5", "F15", "F56", "G5"},
    {"E5", "F35", "F45", "G5"}, {"E5", "F35", "F56", "G5"},
    {"E6", "F13", "F26", "F46"}, {"E6", "F13", "F26", "F56"},
    {"E6", "F13", "F56", "G1"}, {"E6", "F26", "F46", "G6"},
    {"E6", "F26", "F56", "G6"}, {"E6", "F36", "F46", "G6"},
    {"F12", "F14", "F56", "G1"}, {"F12", "F15", "F36", "F46"},
    {"F12", "F25", "F34", "F46"}, {"F12", "F26", "F35", "F45"},
```

```
{"F12", "F26", "F45", "G2"}, {"F13", "F14", "F26", "F56"},
{"F13", "F14", "F56", "G1"}, {"F13", "F16", "F24", "F45"},
{"F13", "F16", "F25", "F45"}, {"F13", "F24", "F36", "F45"},
{"F13", "F25", "F36", "F45"}, {"F13", "F35", "F46", "G3"},
{"F14", "F23", "F35", "F46"}, {"F14", "F25", "F46", "G4"},
{"F15", "F23", "F24", "F56"}, {"F15", "F23", "F34", "F56"},
{"F15", "F24", "F56", "G5"}, {"F15", "F26", "F46", "G6"},
{"F15", "F36", "F46", "G6"}, {"F16", "F23", "F34", "F56"},
{"F16", "F25", "F45", "G5"}, {"F16", "F34", "F56", "G6"},
{"F23", "F26", "F45", "G2"}, {"F23", "F35", "F46", "G3"},
{"F24", "F35", "F56", "G5"}, {"F24", "F36", "F45", "G4"},
{"F25", "F34", "F46", "G4"}, {"E1", "E5", "F12", "F15", "G1"},
{"E1", "E5", "F15", "G1", "G6"}, {"E1", "E5", "F25", "G2", "G4"},
{"E1", "F25", "G2", "G5"}, {"E1", "E6", "F13", "F16", "G1"},
{"E1", "F36", "G3", "G6"}, {"E1", "F13", "F14", "F25", "F36"},
{"E2", "E4", "F23", "F24", "G2"}, {"E2", "E4", "F24", "G2", "G5"},
{"E2", "F34", "G3", "G4"}, {"E2", "E5", "F12", "F25", "G2"},
{"E2", "F15", "G1", "G5"}, {"E2", "F15", "F23", "F26", "F34"},
{"E3", "E4", "F23", "F34", "G3"}, {"E3", "F24", "G2", "G4"},
{"E3", "E4", "F34", "G3", "G6"}, {"E3", "E6", "F13", "F36", "G3"},
{"E3", "E6", "F16", "G1", "G5"}, {"E3", "F16", "G1", "G6"},
{"E3", "E6", "F36", "G3", "G4"}, {"E3", "F13", "F16", "F24", "F35"},
{"E4", "F23", "F24", "F45", "G2"}, {"E4", "F23", "F34", "F46", "G3"},
{"E5", "F12", "F15", "F56", "G1"}, {"E5", "F25", "F45", "G2", "G5"},
{"E6", "F13", "F36", "F46", "G3"}, {"E6", "F16", "F56", "G1", "G6"},
{"F12", "F14", "F25", "F36", "F46"},
{"F12", "F14", "F26", "F35", "F56"}, {"F12", "F15", "F26",
 "F34", "F46"}, {"F13", "F14", "F26", "F35", "F46"},
{"F14", "F23", "F26", "F35", "F45"}, {"F14", "F25", "F36", "F45", "G4"},
{"F15", "F26", "F34", "F56", "G6"}, {"F16", "F23", "F24", "F35", "F56"},
{"F16", "F24", "F35", "F45", "G5"}}];
```