This notebook constructs a fake projective plane. It is a supporting file for the Section 3 of the joint paper of L.B., A.B., and E.F.

In[∘]:=
```
(* to be read if we don't want to recompute points *)
(* The names of the files needs to be
 adjusted according to your situation. *)
fname = "Desktop/FPP2020/SetOfPointsW";
OpenRead[fname];
SetOfPointsW = Read[fname];
Close[fname];

fname = "Desktop/FPP2020/EqsFPPC3W";
OpenRead[fname];
EqsFPPC3W = Read[fname];
Close[fname];
```

In[∘]:=
```
(* useful to have *)
Beautify[x_] := (
   ret = Chop[x, 10^(-Round[Accuracy[x] / 2])];
   root2 = N[√2, Round[Accuracy[x]]];
   ret =
    Rationalize[Re[ret]] + Rationalize[N[Im[ret] / √2, Round[Accuracy[x]]]] ⅈ √2
  );
(* checking *)
tocheck = N[((ⅈ √2 + 565) / 1143), 50]
Beautify[tocheck]
```

Out[∘]= 0.49431321084864391951006124234470691163604549431321 + 0.0012372822068006080916900163816357813460802028656 0 ⅈ

Out[∘]= $\frac{565}{1143} + \frac{i\sqrt{2}}{1143}$

In[•]:=
```
(* I will now find points on the curve W2+W3+W4=0 *)
VarsW = {W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14, W15, W16};
SetOfPointsWwithW234 = {};
prec = 6000;
For[i = 1, i ≤ 100, i++,
  If[Mod[i, 10] == 0, Print[i]];
  rpt1 = (RandomInteger[{1, 100 000}] + I RandomInteger[{1, 100 000}]) / 100 000;
  Eqs = Join[{W1 == 1, W2 + W3 + W4 == 0, W5 + W6 + W7 == rpt1},
    Table[0 == EqsFPPC3W[[j]], {j, 1, 38}]];
  sols = NSolve[Eqs, 200];
  fr = FindRoot[Table[Eqs[[k]], {k, 1, 14}],
    Table[{VarsW[[j]], VarsW[[j]] /. sols[[1]]}, {j, 1, 14}],
    MaxIterations → 100, WorkingPrecision → prec, AccuracyGoal → prec];
  AppendTo[SetOfPointsWwithW234, fr];
 ];
(* quality control *)
For[i = 1, i ≤ 100, i++,
  If[Chop[Sum[Abs[EqsFPPC3W[[j]]] /. SetOfPointsWwithW234[[i]], {j, 1, 38}]] ≠ 0,
    Print["warning, point ", i, " looks fishy"];];];
```

```
10
20
30
40
50
60
70
80
90
100
```

In[•]:=
```
(* Looking for sections of 6K that are zero on W2+W3+W4=0 *)
maxlength = 500;
CoeffList = Table[ToExpression[StringJoin["A", ToString[i]]], {i, 1, maxlength}];

AllMons = {}; For[i = 5, i ≤ 14, i++, For[j = 5, j ≤ i, j++,
  AppendTo[AllMons, VarsW[[i]] × VarsW[[j]]];]];
count = Length[AllMons];
AllMons = Table[
   AllMons[[i]] / (AllMons[[i]] /. Table[VarsW[[k]] → 1, {k, 1, 14}]), {i, 1, count}];

UnivPoly = Sum[AllMons[[k]] × CoeffList[[k]], {k, 1, count}];
Eqs = Table[0 == UnivPoly /. SetOfPointsWwithW234[[k]], {k, 1, count + 20}];
```

```
sols = Chop[ Solve[Eqs][[1]]];
Print["here"];


AllEqs = {};
For[j = 1, j ≤ count, j++,
  UEq = Coefficient[UnivPoly /. sols, CoeffList[[j]]];
  If[ByteCount[UEq] > 100,
   Eq = Block[{$MaxExtraPrecision = 400},
     Sum[Beautify[Coefficient[UEq, AllMons[[n]]]] × AllMons[[n]],
      {n, 1, Length[AllMons]}]];
   AppendTo[AllEqs, Eq];
  ];
 ];
Length[AllEqs]


(* Checking to see which of these
 could be added to the degree two polynomials *)

D4P = {W1^2, W2^2, W3^2, W4^2, W1 W2, W1 W3, W1 W4, W2 W3, W2 W4, W3 W4};
For[j = 1, j ≤ Length[AllEqs], j++,
  AppendTo[D4P, AllEqs[[j]] / (W2 + W3 + W4)];
  Print[j, " ", ByteCount[AllEqs[[j]]], " ",
   MatrixRank[Table[D4P /. SetOfPointsW[[k]], {k, 1, 50}]]];
 ];
D4P = Table[D4P[[k]], {k, 1, 19}];
(* writing into the file *)
fname = "Desktop/FPP2020/Deg4inW";
OpenWrite[fname];
Write[fname, D4P];
Close[fname];
```

here

Out[•]= 28

```
1 36992 11
2 37040 12
3 36896 13
4 38016 14
5 37808 15
6 37936 16
7 37936 17
8 37840 18
9 37856 19
10 37792 19
11 37968 19
12 37872 19
13 37840 19
14 37840 19
15 38048 19
16 37808 19
17 37776 19
18 37776 19
19 37840 19
20 37600 19
21 37568 19
22 36992 19
23 38032 19
24 37904 19
25 37872 19
26 37712 19
27 37664 19
28 37808 19
```

In[•]:=
```
(* finding the conditions of passing through the first two singular points *)
VarsC =
  {c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19};
UnivDeg4 = VarsC.D4P;
pt1 = {W1 → 0, W2 → 1, W3 → 0, W4 → 0, W5 → 1, W6 → 0,
  W7 → 0, W8 → 1, W9 → 0, W10 → 0, W11 → 1, W12 → 0, W13 → 0, W14 → 1}
pt2 = {W1 → 0, W2 → 0, W3 → 1, W4 → 0, W5 → 0, W6 → 1, W7 → 0,
  W8 → 0, W9 → 1, W10 → 0, W11 → 0, W12 → 1, W13 → 0, W14 → 1}
pt3 = {W1 → 0, W2 → 0, W3 → 0, W4 → 1, W5 → 0, W6 → 0, W7 → 1,
  W8 → 0, W9 → 0, W10 → 1, W11 → 0, W12 → 0, W13 → 1, W14 → 1}

EqsOnC = {0 == UnivDeg4 /. pt1, 0 == UnivDeg4 /. pt2};
solsc = Simplify[Solve[EqsOnC]][[1]];
(* tell us which coefficients are solved for *)
Print[Table[solsc[[i]][[1]], {i, 1, Length[solsc]}]];
UnivDeg4throughpt1and2 = UnivDeg4 /. solsc;
```

Out[•]=
```
{W1 → 0, W2 → 1, W3 → 0, W4 → 0, W5 → 1, W6 → 0, W7 → 0,
 W8 → 1, W9 → 0, W10 → 0, W11 → 1, W12 → 0, W13 → 0, W14 → 1}
```

Out[•]=
```
{W1 → 0, W2 → 0, W3 → 1, W4 → 0, W5 → 0, W6 → 1, W7 → 0,
 W8 → 0, W9 → 1, W10 → 0, W11 → 0, W12 → 1, W13 → 0, W14 → 1}
```

Out[•]=
```
{W1 → 0, W2 → 0, W3 → 0, W4 → 1, W5 → 0, W6 → 0, W7 → 1,
 W8 → 0, W9 → 0, W10 → 1, W11 → 0, W12 → 0, W13 → 1, W14 → 1}
```

```
{c2, c3}
```

In[∘]:=
```
(* Need to figure out which equations
 cut out the surface near the singular point *)
rk = 0;
matatpt = {};
indices = {};
For[i = 1, i ≤ 38, i++,
  newrow = N[Table[D[EqsFPPC3W[[i]], VarsW[[k]]] /. pt1, {k, 1, 14}], 500];
  If[MatrixRank[Append[matatpt, newrow]] == rk + 1,
   AppendTo[indices, i];
   AppendTo[matatpt, newrow];
   rk++;
  ];
 ];
indices
Length[indices]
(* so hopefully the first 11 equations do it *)
```

Out[∘]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

Out[∘]= 10

It is important to get rid of the C^* action. We do this by cutting with W14=1.

In[∘]:=
```
VarT = {T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13};
(* W1 = T1, W2 = 1 + T1 T2, ... *)
(* T1 = W1, T2 = (W2-1)/W1 , ... are
 affine coordinates on one of the open sets on the
 cover of blowup of C^14 at (0:1:0:0:1:0:0:1:0:0:1:0:0:1) *)
TtoW = {W1 → T1, W2 → 1 + T2 T1 , W3 → T3 T1 , W4 → T4 T1,
   W5 → 1 + T5 T1 , W6 → T6 T1 , W7 → T7 T1, W8 → 1 + T8 T1 , W9 → T9 T1 ,
   W10 → T10 T1, W11 → 1 + T11 T1 , W12 → T12 T1 , W13 → T13 T1, W14 → 1};
EqsW11 = Table[EqsFPPC3W[[k]], {k, 1, 11}];(* the first 11
 equatons cut out the surface near pt1 *)
(* all equations are divisible by T1, which we remove *)
Expand[(EqsW11 /. TtoW) /. T1 → 0]
RelsT = Table[Sum[SeriesCoefficient[(EqsFPPC3W[[j]] /. TtoW), {T1, 0, k}] T1^(k - 1),
     {k, 1, 3}], {j, 1, 11}];
```

Out[∘]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

In[∘]:=
```
(* coordinates on the exceptional divisor of blowup of the point in C^13*)
VarT2 = {T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13};
solsingeq = Block[{$MaxExtraPrecision = 500},
   Beautify[{1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11} /.
     NSolve[Table[0 == Coefficient[{1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11}.
           RelsT /. T1 → 0, VarT2[[k]]], {k, 1, 12}], 3000][[1]]]];
(* This is a cut that's singular at pt1 *)
singeq = Simplify[solsingeq.EqsW11];
singeqT = Simplify[singeq /. TtoW];

solsexclin = NSolve[Table[0 == RelsT[[j]] /. T1 → 0, {j, 1, 11}],
    {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}, 8000][[1]];
solsexclinexact = {};
For[k = 1, k ≤ 10, k++,
  todo = {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] /. solsexclin;
  AppendTo[solsexclinexact, {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] →
    Beautify[todo /. {T3 → 0, T4 → 0}] +
     Beautify[Coefficient[todo, T3]] T3 + Beautify[Coefficient[todo, T4]] T4];
 ];
solsexclin = solsexclinexact;

(* finding the two exceptional lines *)
sollines =
  Simplify[Solve[SeriesCoefficient[(singeqT /. solsexclin), {T1, 0, 2}] == 0, T3]];
```

In[∘]:=
```
(*******************************************************************************
*******)
(* There is a choice to be made here. A
 wrong choice results in no solutions later on. *)

Line1 = sollines[[1]] (*{T3→1/18 (-79-839 i √2+(3+6 i √2) T4)}*)
Line2 = sollines[[2]] (*{T3→1/99 (753+1520 i √2+(-12-18 i √2) T4)}*)
```

Out[∘]= $\left\{T3 \to \frac{1}{18}\left(-79 - 839\, i \sqrt{2} + \left(3 + 6\, i \sqrt{2}\right) T4\right)\right\}$

Out[∘]= $\left\{T3 \to \frac{1}{99}\left(753 + 1520\, i \sqrt{2} + \left(-12 - 18\, i \sqrt{2}\right) T4\right)\right\}$

In[•]:=
```
(* this was just a sanity check *)
Expand[UnivDeg4throughpt1and2 /. TtoW /. solsexclinexact /. T1 → 0]
N[Expand[(Simplify[(UnivDeg4throughpt1and2 /. TtoW /. solsexclin) / T1] /. T1 → 0) /.
    Line2], 2]
```

Out[•]= 0

Out[•]= $(2.7\times10^2 - 2.7\times10^2\,\mathrm{i})\,c11 - (95. - 1.8\times10^2\,\mathrm{i})\,c12 + (4.2\times10^2 + 2.2\times10^2\,\mathrm{i})\,c13 - (1.8\times10^2 + 71.\,\mathrm{i})\,c14 - (2.0\times10^2 + 3.6\times10^2\,\mathrm{i})\,c15 + (1.1\times10^2 + 2.9\times10^2\,\mathrm{i})\,c16 - (70. - 1.8\times10^2\,\mathrm{i})\,c17 + (78. - 3.1\times10^2\,\mathrm{i})\,c18 - (1.3\times10^2 - 3.8\times10^2\,\mathrm{i})\,c19 + c5 + (7.6 + 22.\,\mathrm{i})\,c8 - (1.2 - 2.0\,\mathrm{i})\,c11\,T4 + (3.3 - 1.0\,\mathrm{i})\,c12\,T4 - (6.2 + 3.1\,\mathrm{i})\,c13\,T4 + (0.0060 - 0.82\,\mathrm{i})\,c14\,T4 + (1.1 + 2.8\,\mathrm{i})\,c15\,T4 - (0.82 + 3.0\,\mathrm{i})\,c16\,T4 + (0.26 - 0.76\,\mathrm{i})\,c17\,T4 - (0.86 - 0.70\,\mathrm{i})\,c18\,T4 + (1.4 - 1.6\,\mathrm{i})\,c19\,T4 - (0.12 + 0.26\,\mathrm{i})\,c8\,T4 + c9\,T4$

In[•]:=
```
(* finding a coordinate chart on the blowup
 so that the complement contains the second line *)
VarsW = {W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14};
W1n = (((T3 /. Line2) /. T4 → 0)) W1 - W3 + (Coefficient[T3 /. Line2, T4]) W4
(* T1 = W1, T2 = (W2-1)/W1 , ... *)
TtoWnew1 =
 FullSimplify[Solve[{W14 == 1, W1n == T1, W2 == 1 + T1 T2, W3 == T1 T3, W4 == T1 T4,
     W5 == 1 + T1 T5, W6 == T1 T6, W7 == T1 T7, W8 == 1 + T1 T8, W9 == T1 T9,
     W10 == T1 T10, W11 == 1 + T1 T11, W12 == T1 T12, W13 == T1 T13}, VarsW][[1]]]
Simplify[(EqsFPPC3W /. TtoWnew1) /. T1 → 0] (* sanity check *)
RelsT =
  Table[Sum[SeriesCoefficient[(EqsFPPC3W[[j]] /. TtoWnew1), {T1, 0, k}] T1^(k-1),
    {k, 1, 4}], {j, 1, 11}];
Series[(singeq /. TtoWnew1), {T1, 0, 1}]
singeqT =
  Sum[SeriesCoefficient[singeq /. TtoWnew1, {T1, 0, k}] T1^{k-2}, {k, 2, 4}];
solsexclin = NSolve[Table[0 == RelsT[[j]] /. T1 → 0, {j, 1, 11}],
    {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}, 8000][[1]];
solsexclinexact = {};
For[k = 1, k ≤ 10, k++,
  todo = {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] /. solsexclin;
  AppendTo[solsexclinexact, {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] →
    Beautify[todo /. {T3 → 0, T4 → 0}] +
     Beautify[Coefficient[todo, T3]] T3 + Beautify[Coefficient[todo, T4]] T4];
 ];
solsexclin = solsexclinexact;
singeqTline = FullSimplify[singeqT /. solsexclin /. T1 → 0];
(* We think of the neighborhood of the line
 as a map from Spec C[y_1][[y_2]] to the blowup. *)
LineThis = Expand[Solve[0 == singeqTline, T3][[1]][[1]]]
```

```
UptoZeroiny2 = Simplify[({T1 → 0, T2 → (T2 /. solsexclin /. LineThis /. T4 → y1),
      T3 → (T3 /. LineThis /. T4 → y1), T4 → y1,
      T5 → (T5 /. solsexclin /. LineThis /. T4 → y1),
      T6 → (T6 /. solsexclin /. LineThis /. T4 → y1),
      T7 → (T7 /. solsexclin /. LineThis /. T4 → y1),
      T8 → (T8 /. solsexclin /. LineThis /. T4 → y1),
      T9 → (T9 /. solsexclin /. LineThis /. T4 → y1),
      T10 → (T10 /. solsexclin /. LineThis /. T4 → y1),
      T11 → (T11 /. solsexclin /. LineThis /. T4 → y1),
      T12 → (T12 /. solsexclin /. LineThis /. T4 → y1),
      T13 → (T13 /. solsexclin /. LineThis /. T4 → y1)
     })];

(* Look at the restriction of the section of O(4H) with condition
 of passing through the two points at the exceptional line. *)
tbz = FullSimplify[
   (Sum[SeriesCoefficient[UnivDeg4throughpt1and2 /. TtoWnew1, {T1, 0, k}] T1^(k - 1),
      {k, 1, 11}]) /. UptoZeroiny2];
Eqs2AB = {0 == tbz /. y1 → 0, 0 == tbz /. y1 → 1};
```

Out[∘]= $\frac{1}{99}\left(753+1520\,i\,\sqrt{2}\right)\,W1-W3+\frac{1}{99}\left(-12-18\,i\,\sqrt{2}\right)\,W4$

Out[∘]= $\left\{W1\to\frac{-99\,i\,T1\,(1+T3)+6\left(-2\,i+3\sqrt{2}\right)\,T1\,T4}{-753\,i+1520\sqrt{2}},\ W2\to 1+T1\,T2,\ W3\to T1\,T3,\right.$
$W4\to T1\,T4,\ W5\to 1+T1\,T5,\ W6\to T1\,T6,\ W7\to T1\,T7,\ W8\to 1+T1\,T8,\ W9\to T1\,T9,$
$\left.W10\to T1\,T10,\ W11\to 1+T1\,T11,\ W12\to T1\,T12,\ W13\to T1\,T13,\ W14\to 1\right\}$

Out[∘]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Out[∘]= $O[T1]^2$

Out[∘]= $T3\to-\frac{20\,775\,007}{27\,881\,577}-\frac{1\,023\,374\,i\,\sqrt{2}}{27\,881\,577}-\frac{3449\,T4}{344\,217}-\frac{21\,025\,i\,\sqrt{2}\,T4}{344\,217}$

In[∘]:=
```
(* now we will write this at the other line,
and use the cyclic symmetry to translate into the knowledge at another point  *)
CyclicW = {W2 → W3, W3 → W4, W4 → W2, W5 → W6, W6 → W7,
   W7 → W5, W8 → W9, W9 → W10, W10 → W8, W11 → W12, W12 → W13, W13 → W11};
(* this was just a sanity check *)
Expand[UnivDeg4throughpt1and2 /. TtoW /. solsexclinexact /. T1 → 0]
(* finding a coordinate chart on the blowup
 so that the complement contains the second line *)
VarsW = {W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14};
W1n = (((T3 /. Line1) /. T4 → 0)) W1 - W3 + (Coefficient[T3 /. Line1, T4]) W4
```

```
(* T1 = W1, T2 = (W2-1)/W1 , ... *)
TtoWnew2 =
 FullSimplify[Solve[{W14 == 1, W1n == T1, W2 == 1 + T1 T2, W3 == T1 T3, W4 == T1 T4,
     W5 == 1 + T1 T5, W6 == T1 T6, W7 == T1 T7, W8 == 1 + T1 T8, W9 == T1 T9,
     W10 == T1 T10, W11 == 1 + T1 T11, W12 == T1 T12, W13 == T1 T13}, VarsW][[1]]]
Simplify[(EqsFPPC3W /. TtoWnew2) /. T1 → 0] (* sanity check *)
RelsT =
  Table[Sum[SeriesCoefficient[(EqsFPPC3W[[j]] /. TtoWnew2), {T1, 0, k}] T1^(k - 1),
    {k, 1, 4}], {j, 1, 11}];
Series[(singeq /. TtoWnew2), {T1, 0, 1}]
singeqT =
  Sum[SeriesCoefficient[singeq /. TtoWnew2, {T1, 0, k}] T1^{k - 2}, {k, 2, 4}];
solsexclin = NSolve[Table[0 == RelsT[[j]] /. T1 → 0, {j, 1, 11}],
    {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}, 8000][[1]];
solsexclinexact = {};
For[k = 1, k ≤ 10, k++,
  todo = {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] /. solsexclin;
  AppendTo[solsexclinexact, {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] →
    Beautify[todo /. {T3 → 0, T4 → 0}] +
     Beautify[Coefficient[todo, T3]] T3 + Beautify[Coefficient[todo, T4]] T4];
 ];
solsexclin = solsexclinexact;
singeqTline = FullSimplify[singeqT /. solsexclin /. T1 → 0];
(* We think of the neighborhood of the line
 as a map from Spec C[y_1][[y_2]] to the blowup. *)
LineThis = Expand[Solve[0 == singeqTline, T3][[1]][[1]]]
UptoZeroiny2 = Simplify[({T1 → 0, T2 → (T2 /. solsexclin /. LineThis /. T4 → y1),
     T3 → (T3 /. LineThis /. T4 → y1), T4 → y1,
     T5 → (T5 /. solsexclin /. LineThis /. T4 → y1),
     T6 → (T6 /. solsexclin /. LineThis /. T4 → y1),
     T7 → (T7 /. solsexclin /. LineThis /. T4 → y1),
     T8 → (T8 /. solsexclin /. LineThis /. T4 → y1),
     T9 → (T9 /. solsexclin /. LineThis /. T4 → y1),
     T10 → (T10 /. solsexclin /. LineThis /. T4 → y1),
     T11 → (T11 /. solsexclin /. LineThis /. T4 → y1),
     T12 → (T12 /. solsexclin /. LineThis /. T4 → y1),
     T13 → (T13 /. solsexclin /. LineThis /. T4 → y1)
    })];

(* Look at the restriction of the section of O(4H) with condition
 of passing through the two points at the exceptional line. *)
tbz = FullSimplify[
   (Sum[SeriesCoefficient[(UnivDeg4throughpt1and2 /. CyclicW /. CyclicW) /.
        TtoWnew2, {T1, 0, k}] T1^(k - 1), {k, 1, 11}]) /. UptoZeroiny2];
```

Out[•]= 0

Out[•]= $\frac{1}{18}\left(-79-839\,i\,\sqrt{2}\right)\,W1-W3+\frac{1}{18}\left(3+6\,i\,\sqrt{2}\right)\,W4$

Out[•]= $\left\{W1\to\frac{3\,T1\,\left(6\,i\,(1+T3)+\left(-i+2\,\sqrt{2}\right)\,T4\right)}{-79\,i+839\,\sqrt{2}},\,W2\to1+T1\,T2,\,W3\to T1\,T3,\right.$
$W4\to T1\,T4,\,W5\to1+T1\,T5,\,W6\to T1\,T6,\,W7\to T1\,T7,\,W8\to1+T1\,T8,\,W9\to T1\,T9,$
$\left.W10\to T1\,T10,\,W11\to1+T1\,T11,\,W12\to T1\,T12,\,W13\to T1\,T13,\,W14\to1\right\}$

Out[•]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Out[•]= $O[T1]^2$

Out[•]= $T3\to-\frac{7\,106\,570}{27\,881\,577}+\frac{1\,023\,374\,i\,\sqrt{2}}{27\,881\,577}-\frac{3449\,T4}{344\,217}-\frac{21\,025\,i\,\sqrt{2}\,T4}{344\,217}$

```
sls = FullSimplify[Solve[Join[Eqs2AB, Eqs2BA]][[1]]];
UnivF = UnivDeg4throughpt1and2 /. sls;
```

We will now write the cubic equation f sigma(f) sigma^2(f) = d^3.

In[∘]:=
```
InvariantC3Deg4 = {};
MatrixValues = {};
rk = 0;
For[i = 1, i ≤ Length[D4P], i++,
  newsym = D4P[[i]] + (D4P[[i]] /. CyclicW) + (D4P[[i]] /. CyclicW /. CyclicW);
  newrow = Block[{$MaxExtraPrecision = 2000},
    Table[newsym /. SetOfPointsW[[k]], {k, 1, Length[SetOfPointsW]}]];
  If[MatrixRank[Append[MatrixValues, newrow]] == rk + 1,
   rk++;
   Print[{i, rk}];
   AppendTo[MatrixValues, newrow];
   AppendTo[InvariantC3Deg4, newsym];
  ];
 ];
VarsD = {d1, d2, d3, d4, d5, d6, d7};
sls = Simplify[Solve[0 == VarsD.InvariantC3Deg4 /. pt1][[1]]];
UnivD = VarsD.InvariantC3Deg4 /. sls;
VarsDleft = Complement[VarsD, {d7}];
VarsCleft = {};
VarsC =
  {c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19};
For[i = 1, i ≤ 19, i++,
  If[ByteCount[Coefficient[UnivF, VarsC[[i]]]] > 30,
    AppendTo[VarsCleft, VarsC[[i]]]];
 ];
VarsCleft
Length[VarsCleft]
```

```
{1, 1}
{2, 2}
{5, 3}
{8, 4}
{11, 5}
{12, 6}
{13, 7}
```

Out[∘]= {c1, c4, c7, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19}

Out[∘]= 13

In[∘]:=
```
AllMonsCD = {};
For[i = 1, i ≤ 13, i++, For[j = 1, j ≤ i, j++, For[k = 1, k ≤ j, k++,
    AppendTo[AllMonsCD, VarsCleft[[i]] × VarsCleft[[j]] × VarsCleft[[k]]];
   ]]];
For[i = 1, i ≤ 6, i++, For[j = 1, j ≤ i, j++, For[k = 1, k ≤ j, k++,
    AppendTo[AllMonsCD, VarsDleft[[i]] × VarsDleft[[j]] × VarsDleft[[k]]];
   ]]];
Length[AllMonsCD]
```

Out[∘]=
```
511
```

Need to work with longer expansions. Computing points takes awhile.

In[∘]:=
```
SetOfPointsWextra = {};
prec = 60 000;
For[i = 1, i ≤ 100, i++,
  If[Mod[i, 20] == 2, Print[i]];
  rpt1 = (RandomInteger[{1, 100 000}] + I RandomInteger[{1, 100 000}]) / 100 000;
  rpt2 = (RandomInteger[{1, 100 000}] + I RandomInteger[{1, 100 000}]) / 100 000;
  Eqs = Join[{W1 == 1, W2 + W3 + W4 == rpt1, W5 + W6 + W7 == rpt2},
    Table[0 == EqsFPPC3W[[j]], {j, 1, 38}]];
  sols = NSolve[Eqs, 200];
  fr = FindRoot[Table[Eqs[[k]], {k, 1, 14}],
    Table[{VarsW[[j]], VarsW[[j]] /. sols[[1]]}, {j, 1, 14}],
    MaxIterations → 200, WorkingPrecision → prec, AccuracyGoal → prec];
  AppendTo[SetOfPointsWextra, fr];
 ];
```

```
2
22
42
62
82
```

```
In[•]:= Equations = {};
For[i = 1, i ≤ Length[SetOfPointsWextra], i++,
  If[Mod[i, 50] == 5, Print[i]];
  valueatpt = Block[{$MaxExtraPrecision = 60 000},
    (UnivF (UnivF /. CyclicW) (UnivF /. CyclicW /. CyclicW) - UnivD^3) /.
     SetOfPointsWextra[[i]]];
  Block[{$MaxExtraPrecision = 60 000}, AppendTo[Equations,
    Table[Coefficient[valueatpt, AllMonsCD[[k]]], {k, 1, 511}]]];
 ];
Print["Go have dinner."];
rrEqs = RowReduce[Chop[Equations, 10^(-50 000)]];

N: Internal precision limit $MaxExtraPrecision = 60000.` reached while evaluating
  (−≪60049≫ + ≪60048≫ i) (864473855140782959457547504218824588640946664272762029934018 8⋱
       002805914098969433206913777045964372161068370213751065579288887800909085564902847
       7083287934639169671121608941278368185634817969167876778580743797014396061 ≪49≫ 858
       309322137774935696065163622628438944774762215248532461658236107552780079122211⋱
       66442243376488705044636995258678206182490354434424436806029540447634468385 6⋱
       34772415237548440058956196650675645428838759341180345884314 + ≪1≫).

N: Internal precision limit $MaxExtraPrecision = 60000.` reached while evaluating
  (−≪60045≫ + ≪60046≫ i) (−714188130705721139262542106548837083010466058857646872624172 3⋱
         193014448092467389998187664527189955492683676403566518755048643067712470125 82⋱
         500845116404829564820645261275721593531179309541456194604494520487788777061 ≪52
         ≫ 02700416788106780685331607315693962057804272851558918713350094698733107988⋱
         228588826745551807354134014578051417784003146380515303978554475522088012270 3⋱
         183490009634746816516117385964483194772957794494167141656954218 1 + ≪1≫).

N: Internal precision limit $MaxExtraPrecision = 60000.` reached while evaluating
  (−≪60049≫ − ≪60048≫ i) (864473855140782959457547504218824588640946664272762029934018 8⋱
       002805914098969433206913777045964372161068370213751065579288887800909085564902847
       7083287934639169671121608941278368185634817969167876778580743797014396061 ≪49≫ 858
       309322137774935696065163622628438944774762215248532461658236107552780079122211⋱
       66442243376488705044636995258678206182490354434424436806029540447634468385 6⋱
       34772415237548440058956196650675645428838759341180345884314 + ≪1≫).

General: Further output of N::meprec will be suppressed during this calculation.

5

55

Go have dinner.
```

In[•]:=
```
(* There is a way to improve on speed by prescaling *)
den = 1;
For[numtry = 1, numtry < 1000, numtry++,
  k = RandomInteger[510] + 1; j = RandomInteger[57] + 1;
  denmore = Denominator[
    Block[{$MaxExtraPrecision = 10 000}, Rationalize[den Re[rrEqs[[j]][[k]]]]]];
  den = denmore den;
 ];
N[den]
```

Out[•]= $1.314335234490630 \times 10^{14721}$

In[•]:=
```
(* used that denominator for all  entries, it speeds up things greatly *)
rrEqsExact = {};
For[i = 1, i ≤ 58, i++,
  If[Mod[i, 20] == 4, Print[i]];
  Block[{$MaxExtraPrecision = 10 000},
   AppendTo[rrEqsExact, Beautify[ den rrEqs[[i]]]]]];
 ];
Table[Accuracy[rrEqsExact[[i]]], {i, 1, 58}]
```

4

24

44

Out[•]= {∞, ∞,
∞, ∞}

In[∘]:=
```
Eqs58 = {};
For[k = 1, k ≤ 58, k++,
  runningLCM = 1;
  eq = rrEqsExact[[k]];
  For[i = 1, i ≤ 511, i++,
   a = rrEqsExact[[k]][[i]];
   b = LCM[Denominator[Re[a]], Denominator[Im[a] / 2^(1/2)]];
   runningLCM = LCM[runningLCM, b];
  ];
  AppendTo[Eqs58, Expand[runningLCM eq]];
 ];
Eqs58withrr = {};
For[k = 1, k ≤ 58, k++,
  AppendTo[Eqs58withrr,
    Table[Re[Eqs58[[k]][[i]]] + rr Im[Eqs58[[k]][[i]] / 2^(1/2)], {i, 1, 511}]
   ];
 ];
(* sanity check *)
DeleteDuplicates[Flatten[(Eqs58withrr /. rr → (-2)^(1/2)) - Eqs58]]
```

Out[∘]= {0}

In[∘]:=
```
The58Eqswithrr = Table[Eqs58withrr[[k]].AllMonsCD, {k, 1, 58}];
```

Let us also add the linear equations coming from the C3 fixed points.

In[∘]:=
```
w = Exp[2 Pi I / 3]
eqsfixed = Append[EqsFPPC3W, W1 - 1];
eqsfixed = Join[eqsfixed, Table[(VarsW /. CyclicW)[[k]] - VarsW[[k]], {k, 1, 14}]];
slsfixed = NSolve[Table[0 == eqsfixed[[k]], {k, 1, Length[eqsfixed]}], 10000];
slsfixed = Chop[slsfixed, 10^(-5000)];
Length[slsfixed]
N[VarsW /. slsfixed[[4]]]
N[VarsW /. slsfixed[[5]]]
slsfixedexact1 =
 Block[{$MaxExtraPrecision = 1000}, RootApproximant[VarsW /. slsfixed[[1]], 4]]
slsfixedexact2 = Block[{$MaxExtraPrecision = 1000},
  RootApproximant[VarsW /. slsfixed[[2]], 4]]
N[D4P[[2]] /. slsfixed] (* we see that these are the three points *)

EqLin1 = Block[{$MaxExtraPrecision = 10000},
   Numerator[Factor[Sum[RootApproximant[Coefficient[UnivF, VarsCleft[[k]]] /.
          slsfixed[[1]], 2] × VarsCleft[[k]], {k, 1, 13}] -
```

```
      Sum[RootApproximant[Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[1]], 2] ×
        VarsDleft[[k]], {k, 1, 6}]
    ]]];
EqLin1withrr = Sum[
    (Re[Coefficient[EqLin1, VarsCleft[[k]]]] +
       rr Im[Coefficient[EqLin1, VarsCleft[[k]]] / 2^(1/2)])
     VarsCleft[[k]], {k, 1, 13}] + Sum[
    (Re[Coefficient[EqLin1, VarsDleft[[k]]]] +
       rr Im[Coefficient[EqLin1, VarsDleft[[k]]] / 2^(1/2)])
     VarsDleft[[k]], {k, 1, 6}];

(* this tells us what to do *)
Block[{$MaxExtraPrecision = 10000},
 RootApproximant[(D4P[[2]] /. slsfixed[[5]]) N[w, 10000] +
   (D4P[[2]] /. slsfixed[[4]]) N[w^2, 10000], 2]]
Block[{$MaxExtraPrecision = 10000}, RootApproximant[
  (D4P[[2]] /. slsfixed[[5]]) N[w^2, 10000] +
   (D4P[[2]] /. slsfixed[[4]]) N[w, 10000], 2]]

EqLin2 =
  Block[{$MaxExtraPrecision = 10000},
   Numerator[Factor[
     Sum[RootApproximant[
         N[w, 10000] (Coefficient[UnivF, VarsCleft[[k]]] /. slsfixed[[4]])
          + N[w^2, 10000] (Coefficient[UnivF, VarsCleft[[k]]] /. slsfixed[[5]])
         , 2] × VarsCleft[[k]], {k, 1, 13}]
      -
      Sum[RootApproximant[(
          N[w^2, 10000] (Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[4]])
           + N[w, 10000] (Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[5]])
         ), 2] × VarsDleft[[k]], {k, 1, 6}]
    ]]
  ];

EqLin2withrr = Sum[
    (Re[Coefficient[EqLin2, VarsCleft[[k]]]] +
       rr Im[Coefficient[EqLin2, VarsCleft[[k]]] / 2^(1/2)])
     VarsCleft[[k]], {k, 1, 13}] + Sum[
    (Re[Coefficient[EqLin2, VarsDleft[[k]]]] +
       rr Im[Coefficient[EqLin2, VarsDleft[[k]]] / 2^(1/2)])
     VarsDleft[[k]], {k, 1, 6}];

EqLin3 =
  Block[{$MaxExtraPrecision = 10000},
```

```
    Numerator[Factor[
      Sum[RootApproximant[
          N[w^2, 10 000] (Coefficient[UnivF, VarsCleft[[k]]] /. slsfixed[[4]])
           + N[w, 10 000] (Coefficient[UnivF, VarsCleft[[k]]] /. slsfixed[[5]])
          , 2] × VarsCleft[[k]], {k, 1, 13}]
       -
       Sum[RootApproximant[(
           N[1, 10 000] (Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[4]])
            + N[1, 10 000] (Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[5]])
          ), 2] × VarsDleft[[k]], {k, 1, 6}]
     ]]
   ];

EqLin3withrr = Sum[
     (Re[Coefficient[EqLin3, VarsCleft[[k]]]] +
        rr Im[Coefficient[EqLin3, VarsCleft[[k]]] / 2^(1 / 2)])
      VarsCleft[[k]], {k, 1, 13}] + Sum[
     (Re[Coefficient[EqLin3, VarsDleft[[k]]]] +
        rr Im[Coefficient[EqLin3, VarsDleft[[k]]] / 2^(1 / 2)])
      VarsDleft[[k]], {k, 1, 6}];

EqLin4 =
   Block[{$MaxExtraPrecision = 10 000},
    Numerator[Factor[
      Sum[RootApproximant[
          N[w, 10 000] (Coefficient[UnivF, VarsCleft[[k]]] /. slsfixed[[4]])
           + N[w^2, 10 000] (Coefficient[UnivF, VarsCleft[[k]]] /. slsfixed[[5]])
          , 2] × VarsCleft[[k]], {k, 1, 13}]
       -
       Sum[RootApproximant[(
           N[1, 10 000] (Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[4]])
            + N[1, 10 000] (Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[5]])
          ), 2] × VarsDleft[[k]], {k, 1, 6}]
     ]]
   ];

EqLin4withrr = Sum[
     (Re[Coefficient[EqLin4, VarsCleft[[k]]]] +
        rr Im[Coefficient[EqLin4, VarsCleft[[k]]] / 2^(1 / 2)])
      VarsCleft[[k]], {k, 1, 13}] + Sum[
     (Re[Coefficient[EqLin4, VarsDleft[[k]]]] +
        rr Im[Coefficient[EqLin4, VarsDleft[[k]]] / 2^(1 / 2)])
      VarsDleft[[k]], {k, 1, 6}];
EqLin5 =
```

```
Block[{$MaxExtraPrecision = 10 000},
 Numerator[Factor[
   Sum[RootApproximant[
      N[w^2, 10 000] (Coefficient[UnivF, VarsCleft[[k]]] /. slsfixed[[4]])
       + N[w, 10 000] (Coefficient[UnivF, VarsCleft[[k]]] /. slsfixed[[5]])
      , 2] × VarsCleft[[k]], {k, 1, 13}]
    -
    Sum[RootApproximant[(
       N[w, 10 000] (Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[4]])
        + N[w^2, 10 000] (Coefficient[UnivD, VarsDleft[[k]]] /. slsfixed[[5]])
      ), 2] × VarsDleft[[k]], {k, 1, 6}]
  ]]
];

EqLin5withrr = Sum[
   (Re[Coefficient[EqLin5, VarsCleft[[k]]]] +
      rr Im[Coefficient[EqLin5, VarsCleft[[k]]] / 2^(1/2)])
    VarsCleft[[k]], {k, 1, 13}] + Sum[
   (Re[Coefficient[EqLin5, VarsDleft[[k]]]] + rr Im[
        Coefficient[EqLin5, VarsDleft[[k]]] / 2^(1/2)]) VarsDleft[[k]], {k, 1, 6}];
```

Out[•]= $e^{\frac{2 i \pi}{3}}$

Out[•]= 6

Out[•]= {1., -35.8939 + 24.5636 i, -35.8939 + 24.5636 i, -35.8939 + 24.5636 i,
 -646.798 - 279.746 i, -646.798 - 279.746 i, -646.798 - 279.746 i, -933.819 - 812.972 i,
 -933.819 - 812.972 i, -933.819 - 812.972 i, -745.356 - 490.895 i,
 -745.356 - 490.895 i, -745.356 - 490.895 i, -1536.44 - 239.685 i}

Out[•]= {1., 22.8939 + 0.892242 i, 22.8939 + 0.892242 i, 22.8939 + 0.892242 i,
 -186.41 + 194.055 i, -186.41 + 194.055 i, -186.41 + 194.055 i,
 217.035 - 137.037 i, 217.035 - 137.037 i, 217.035 - 137.037 i,
 40.71 + 92.2222 i, 40.71 + 92.2222 i, 40.71 + 92.2222 i, -374.704 + 576.302 i}

Out[•]= $\left\{1, \frac{1}{2}\left(26 - 35 i \sqrt{2}\right), \frac{1}{2}\left(26 - 35 i \sqrt{2}\right), \frac{1}{2}\left(26 - 35 i \sqrt{2}\right),\right.$ -410.… + 132.… i,
 -410.… + 132.… i, -410.… + 132.… i, -142.… - 73.3… i,
 -142.… - 73.3… i, -142.… - 73.3… i, -131.… + 133.… i,
 -131.… + 133.… i, -131.… + 133.… i, -265.… + 778.… i}

Out[∘]= $\left\{1, \frac{1}{2}\left(26-35\, i \sqrt{2}\right), \frac{1}{2}\left(26-35\, i \sqrt{2}\right), \frac{1}{2}\left(26-35\, i \sqrt{2}\right),\right.$
410.… - 132.… i, 410.… - 132.… i, 410.… - 132.… i,
142.… + 73.3… i, 142.… + 73.3… i, 142.… + 73.3… i,
131.… - 133.… i, 131.… - 133.… i, 131.… - 133.… i, 265.… - 778.… i}

Out[∘]= {-443.5 - 643.467 i, -443.5 - 643.467 i, 685. - 1763.37 i,
685. - 1763.37 i, 523.334 + 40.8537 i, 523.334 + 40.8537 i}

Out[∘]= $\frac{1}{3}\left(-6500+1530\, i \sqrt{2}\right)$

Out[∘]= $\frac{1}{3}\left(2875+2124\, i \sqrt{2}\right)$

In[∘]:=
```
ByteCount[The58Eqswithrr]
VarsCDleft = Join[VarsCleft, VarsDleft]
Length[VarsCDleft]
Eqs58mod19 = Mod[Eqs58withrr /. rr → 6, 19];
EqLin1mod19 = Sum[VarsCDleft[[k]]
    Mod[Coefficient[EqLin1withrr, VarsCDleft[[k]]] /. rr → 6, 19], {k, 1, 19}];
EqLin2mod19 = Sum[VarsCDleft[[k]]
    Mod[Coefficient[EqLin2withrr, VarsCDleft[[k]]] /. rr → 6, 19], {k, 1, 19}];
EqLin3mod19 = Sum[VarsCDleft[[k]]
    Mod[Coefficient[EqLin3withrr, VarsCDleft[[k]]] /. rr → 6, 19], {k, 1, 19}];
EqLin4mod19 = Sum[VarsCDleft[[k]]
    Mod[Coefficient[EqLin4withrr, VarsCDleft[[k]]] /. rr → 6, 19], {k, 1, 19}];
EqLin5mod19 = Sum[VarsCDleft[[k]]
    Mod[Coefficient[EqLin5withrr, VarsCDleft[[k]]] /. rr → 6, 19], {k, 1, 19}];
```

Out[∘]= 318 776 744

Out[∘]= {c1, c4, c7, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19, d1, d2, d3, d4, d5, d6}

Out[∘]= 19

I am looking at the second order neighborhood of the exceptional line to write quadratic equations.

In[∘]:=
```
(* this was just a sanity check *)
Expand[UnivDeg4throughpt1and2 /. TtoW /. solsexclinexact /. T1 → 0]
(* finding a coordinate chart on the blowup
 so that the complement contains the second line *)
VarsW = {W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14};
W1n = (((T3 /. Line2) /. T4 → 0)) W1 - W3 + (Coefficient[T3 /. Line2, T4]) W4
(* T1 = W1, T2 = (W2-1)/W1 , ... *)
```

```
TtoWnew1 =
 FullSimplify[Solve[{W14 == 1, W1n == T1, W2 == 1 + T1 T2, W3 == T1 T3, W4 == T1 T4,
     W5 == 1 + T1 T5, W6 == T1 T6, W7 == T1 T7, W8 == 1 + T1 T8, W9 == T1 T9,
     W10 == T1 T10, W11 == 1 + T1 T11, W12 == T1 T12, W13 == T1 T13}, VarsW][[1]]]
Simplify[(EqsFPPC3W /. TtoWnew1) /. T1 → 0] (* sanity check *)
RelsT =
  Table[Sum[SeriesCoefficient[(EqsFPPC3W[[j]] /. TtoWnew1), {T1, 0, k}] T1^(k - 1),
    {k, 1, 7}], {j, 1, 11}];
Series[(singeq /. TtoWnew1), {T1, 0, 1}]
singeqT =
  Sum[SeriesCoefficient[singeq /. TtoWnew1, {T1, 0, k}] T1^{k - 2}, {k, 2, 8}];
solsexclin = NSolve[Table[0 == RelsT[[j]] /. T1 → 0, {j, 1, 11}],
    {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}, 8000][[1]];
solsexclinexact = {};
For[k = 1, k ≤ 10, k++,
  todo = {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] /. solsexclin;
  AppendTo[solsexclinexact, {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] →
    Beautify[todo /. {T3 → 0, T4 → 0}] +
     Beautify[Coefficient[todo, T3]] T3 + Beautify[Coefficient[todo, T4]] T4];
 ];
solsexclin = solsexclinexact;
singeqTline = FullSimplify[singeqT /. solsexclin /. T1 → 0];
(* We think of the neighborhood of the line
 as a map from Spec C[y_1][[y_2]] to the blowup. *)
LineThis = Expand[Solve[0 == singeqTline, T3][[1]][[1]]]
UptoZeroiny2 = Simplify[({T1 → 0, T2 → (T2 /. solsexclin /. LineThis /. T4 → y1),
     T3 → (T3 /. LineThis /. T4 → y1), T4 → y1,
     T5 → (T5 /. solsexclin /. LineThis /. T4 → y1),
     T6 → (T6 /. solsexclin /. LineThis /. T4 → y1),
     T7 → (T7 /. solsexclin /. LineThis /. T4 → y1),
     T8 → (T8 /. solsexclin /. LineThis /. T4 → y1),
     T9 → (T9 /. solsexclin /. LineThis /. T4 → y1),
     T10 → (T10 /. solsexclin /. LineThis /. T4 → y1),
     T11 → (T11 /. solsexclin /. LineThis /. T4 → y1),
     T12 → (T12 /. solsexclin /. LineThis /. T4 → y1),
     T13 → (T13 /. solsexclin /. LineThis /. T4 → y1)
    })];

RelsTplus = Append[RelsT, singeqT[[1]]];

param = UptoZeroiny2;
param1 = {T1 → y2, T2 → (T2 /. param) + t2 y2, T3 → (T3 /. param) + t3 y2, T4 → y1,
   T5 → (T5 /. param) + t5 y2,
   T6 → (T6 /. param) + t6 y2,
```

```
    T7 → (T7 /. param) + t7 y2,
    T8 → (T8 /. param) + t8 y2,
    T9 → (T9 /. param) + t9 y2,
    T10 → (T10 /. param) + t10 y2,
    T11 → (T11 /. param) + t11 y2,
    T12 → (T12 /. param) + t12 y2,
    T13 → (T13 /. param) + t13 y2};

solvenext =
  Simplify[Solve[Table[0 == SeriesCoefficient[RelsTplus[[k]] /. param1, {y2, 0, 1}],
      {k, 1, 12}], {t2, t3, t5, t6, t7, t8, t9, t10, t11, t12, t13}][[1]]];
param1 = param1 /. solvenext;

UnivFonLine1 =
  Block[{$MaxExtraPrecision = 10 000}, Sum[Beautify[Table[SeriesCoefficient[
        N[Coefficient[UnivF, VarsCleft[[j]]] /. TtoWnew1 /. param1, 20 000],
        {y1, 0, k}, {y2, 0, 2}], {k, 0, 3}]] × VarsCleft[[j]], {j, 1, 13}]];

dens = Denominator[Factor[UnivFonLine1]];
den = LCM[LCM[dens[[1]], dens[[2]]], LCM[dens[[3]], dens[[4]]]]
UnivFonLine1 = Simplify[UnivFonLine1 den];
```

Out[∘]= 0

Out[∘]= $\frac{1}{99}\left(753+1520\,i\,\sqrt{2}\right)\,W1-W3+\frac{1}{99}\left(-12-18\,i\,\sqrt{2}\right)\,W4$

Out[∘]= $\left\{W1\to\frac{-99\,i\,T1\,(1+T3)+6\left(-2\,i+3\,\sqrt{2}\right)\,T1\,T4}{-753\,i+1520\,\sqrt{2}},\ W2\to 1+T1\,T2,\ W3\to T1\,T3,\ W4\to T1\,T4,\ W5\to 1+T1\,T5,\ W6\to T1\,T6,\ W7\to T1\,T7,\ W8\to 1+T1\,T8,\ W9\to T1\,T9,\ W10\to T1\,T10,\ W11\to 1+T1\,T11,\ W12\to T1\,T12,\ W13\to T1\,T13,\ W14\to 1\right\}$

Out[∘]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Out[∘]= $O[T1]^2$

Out[∘]= $T3\to-\frac{20\,775\,007}{27\,881\,577}-\frac{1\,023\,374\,i\,\sqrt{2}}{27\,881\,577}-\frac{3449\,T4}{344\,217}-\frac{21\,025\,i\,\sqrt{2}\,T4}{344\,217}$

Out[∘]= 62 019 081 634 971 993 761 420 620 663 324 249 405 278 066 009 791 408 653 778 891 395 892 614 ⋱
390 098 038 710 984 343 696 218 978 766 412 815 950 005 386 553 839 421 355 581 588 282 820 524 ⋱
078 377 703 578 389 474 972 829 359 780 526 979 657 916 783 619 891 526 889 679 342 242 843 924 ⋱
732 145 277 626 043 297 920 078 711 696 590 891 575 798 505 434 939 373 973 041 909 947 819 470 ⋱
065 094 926 461 485 127 853 410 156 771 711 268 817 259 776 774 244 006 439 315 957 609 123 898 ⋱
696 398 364 655 411 948 457 682 618 295 200 981 279 821 968 861 784 991 128 055 018 543 290 636 ⋱
662 565 908 758 273 383 361 550 530 434 106 889 369 387 452 155 852 668 120 249 109 453 182

In[∘]:=
```
UnivDonLine1 =
  Block[{$MaxExtraPrecision = 10 000}, Sum[Beautify[Table[SeriesCoefficient[
        N[Coefficient[UnivD, VarsDleft[[j]]] /. TtoWnew1 /. param1, 20 000],
        {y1, 0, k}, {y2, 0, 1}], {k, 0, 3}]] × VarsDleft[[j]], {j, 1, 6}]];
dens = Denominator[Factor[UnivDonLine1]];
den = LCM[dens[[1]], dens[[2]]]
UnivDonLine1 = Simplify[UnivDonLine1 den];
```

Out[∘]= 1 210 594 593 397 286 870 492 273 714 461 297 133 849 103 171 031 660 848 941 357 995 877 355 898 ⋱
521 231 769 552 005 938 743 132 618 945 203 946 201 134 534 854 579 372 621 027 957 381 720 485 ⋱
619 246 980 526 001 983 479 455 097 852 694 445 046 942 963 271 937 618 829 769 216 134 244 194 ⋱
555 528 962 321 042 205 021 252 349 935 294 782 912 589 920 333 116 292 075 215 528 744 223 772 ⋱
229 626 236 922 093 750 608 817 736 473 450 395 521 300 363 554 442 609 388 542 652 504 559 585 ⋱
638 593 468 359 347 312 730 133 164 322 331 949 488 067 082 112 962 968 297 359 564 504 461 325 ⋱
492 805 876 654 239 483

Setting up six quadratic equations

In[∘]:=
```
{aa0, aa1, aa2, aa3} = UnivFonLine1;
{bb0, bb1} = {UnivDonLine1[[1]], UnivDonLine1[[2]]};
QuadEqs = {3 aa3 bb0 - aa2 bb1, aa2 bb0 - aa1 bb1, aa1 bb0 - 3 aa0 bb1,
   9 aa0 aa3 - aa1 aa2, 3 aa0 aa2 - aa1^2, 3 aa1 aa3 - aa2^2};
```

In[∘]:=
```
(* now the other line *)
(* finding a coordinate chart on the blowup
 so that the complement contains the first line *)
VarsW = {W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14};
W1n = (((T3 /. Line1) /. T4 → 0)) W1 - W3 + (Coefficient[T3 /. Line1, T4]) W4
(* T1 = W1, T2 = (W2-1)/W1 , ... *)
TtoWnew1 =
 FullSimplify[Solve[{W14 == 1, W1n == T1, W2 == 1 + T1 T2, W3 == T1 T3, W4 == T1 T4,
     W5 == 1 + T1 T5, W6 == T1 T6, W7 == T1 T7, W8 == 1 + T1 T8, W9 == T1 T9,
     W10 == T1 T10, W11 == 1 + T1 T11, W12 == T1 T12, W13 == T1 T13}, VarsW][[1]]]
Simplify[(EqsFPPC3W /. TtoWnew1) /. T1 → 0] (* sanity check *)
RelsT =
  Table[Sum[SeriesCoefficient[(EqsFPPC3W[[j]] /. TtoWnew1), {T1, 0, k}] T1^(k - 1),
    {k, 1, 4}], {j, 1, 11}];
Series[(singeq /. TtoWnew1), {T1, 0, 1}]
```

```
singeqT =
  Sum[SeriesCoefficient[singeq /. TtoWnew1, {T1, 0, k}] T1^{k-2}, {k, 2, 5}];
solsexclin = NSolve[Table[0 == RelsT[[j]] /. T1 → 0, {j, 1, 11}],
    {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}, 8000][[1]];
solsexclinexact = {};
For[k = 1, k ≤ 10, k++,
  todo = {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] /. solsexclin;
  AppendTo[solsexclinexact, {T2, T5, T6, T7, T8, T9, T10, T11, T12, T13}[[k]] →
    Beautify[todo /. {T3 → 0, T4 → 0}] +
     Beautify[Coefficient[todo, T3]] T3 + Beautify[Coefficient[todo, T4]] T4];
 ];
solsexclin = solsexclinexact;
singeqTline = FullSimplify[singeqT /. solsexclin /. T1 → 0];
(* We think of the neighborhood of the line
 as a map from Spec C[y_1][[y_2]] to the blowup. *)
LineThis = Expand[Solve[0 == singeqTline, T3][[1]][[1]]]
UptoZeroiny2 = Simplify[({T1 → 0, T2 → (T2 /. solsexclin /. LineThis /. T4 → y1),
     T3 → (T3 /. LineThis /. T4 → y1), T4 → y1,
     T5 → (T5 /. solsexclin /. LineThis /. T4 → y1),
     T6 → (T6 /. solsexclin /. LineThis /. T4 → y1),
     T7 → (T7 /. solsexclin /. LineThis /. T4 → y1),
     T8 → (T8 /. solsexclin /. LineThis /. T4 → y1),
     T9 → (T9 /. solsexclin /. LineThis /. T4 → y1),
     T10 → (T10 /. solsexclin /. LineThis /. T4 → y1),
     T11 → (T11 /. solsexclin /. LineThis /. T4 → y1),
     T12 → (T12 /. solsexclin /. LineThis /. T4 → y1),
     T13 → (T13 /. solsexclin /. LineThis /. T4 → y1)
    })];

RelsTplus = Append[RelsT, singeqT[[1]]];
param = UptoZeroiny2;
param1 = {T1 → y2, T2 → (T2 /. param) + t2 y2, T3 → (T3 /. param) + t3 y2, T4 → y1,
   T5 → (T5 /. param) + t5 y2,
   T6 → (T6 /. param) + t6 y2,
   T7 → (T7 /. param) + t7 y2,
   T8 → (T8 /. param) + t8 y2,
   T9 → (T9 /. param) + t9 y2,
   T10 → (T10 /. param) + t10 y2,
   T11 → (T11 /. param) + t11 y2,
   T12 → (T12 /. param) + t12 y2,
   T13 → (T13 /. param) + t13 y2};

solvenext =
  Simplify[Solve[Table[0 == SeriesCoefficient[RelsTplus[[k]] /. param1, {y2, 0, 1}],
```

```
      {k, 1, 12}], {t2, t3, t5, t6, t7, t8, t9, t10, t11, t12, t13}][[1]]];
param1 = param1 /. solvenext;

UnivFonLine2 = Block[{$MaxExtraPrecision = 10 000},
   Sum[Beautify[Table[SeriesCoefficient[N[Coefficient[UnivF /. CyclicW /. CyclicW,
            VarsCleft[[j]]] /. TtoWnew1 /. param1, 20 000],
        {y1, 0, k}, {y2, 0, 2}], {k, 0, 3}]] × VarsCleft[[j]], {j, 1, 13}]];
dens = Denominator[Factor[UnivFonLine2]];
den = LCM[LCM[dens[[1]], dens[[2]]], LCM[dens[[3]], dens[[4]]]]
UnivFonLine2 = Simplify[UnivFonLine2 den];
UnivDonLine2 =
  Block[{$MaxExtraPrecision = 10 000}, Sum[Beautify[Table[SeriesCoefficient[
        N[Coefficient[UnivD, VarsDleft[[j]]] /. TtoWnew1 /. param1, 20 000],
        {y1, 0, k}, {y2, 0, 1}], {k, 0, 3}]] × VarsDleft[[j]], {j, 1, 6}]];
{Simplify[UnivDonLine2[[3]]], Simplify[UnivDonLine2[[4]]]}
dens = Denominator[Factor[UnivDonLine2]];
den = LCM[dens[[1]], dens[[2]]]
UnivDonLine2 = Simplify[UnivDonLine2 den];
{aa0, aa1, aa2, aa3} = UnivFonLine2;
{bb0, bb1} = {UnivDonLine2[[1]], UnivDonLine2[[2]]};
QuadEqs = Join[QuadEqs, {3 aa3 bb0 - aa2 bb1, aa2 bb0 - aa1 bb1,
    aa1 bb0 - 3 aa0 bb1, 9 aa0 aa3 - aa1 aa2, 3 aa0 aa2 - aa1^2, 3 aa1 aa3 - aa2^2}];
```

Out[∘]= $\frac{1}{18}\left(-79-839\,i\,\sqrt{2}\right)\,W1-W3+\frac{1}{18}\left(3+6\,i\,\sqrt{2}\right)\,W4$

Out[∘]= $\left\{W1\to\frac{3\,T1\,\left(6\,i\,(1+T3)+\left(-i+2\sqrt{2}\right)\,T4\right)}{-79\,i+839\sqrt{2}},\ W2\to 1+T1\,T2,\ W3\to T1\,T3,\right.$
$W4\to T1\,T4,\ W5\to 1+T1\,T5,\ W6\to T1\,T6,\ W7\to T1\,T7,\ W8\to 1+T1\,T8,\ W9\to T1\,T9,$
$\left.W10\to T1\,T10,\ W11\to 1+T1\,T11,\ W12\to T1\,T12,\ W13\to T1\,T13,\ W14\to 1\right\}$

Out[∘]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Out[∘]= $O[T1]^2$

Out[∘]= $T3\to-\frac{7\,106\,570}{27\,881\,577}+\frac{1\,023\,374\,i\,\sqrt{2}}{27\,881\,577}-\frac{3449\,T4}{344\,217}-\frac{21\,025\,i\,\sqrt{2}\,T4}{344\,217}$

Out[◦]= 5 638 098 330 451 999 432 856 420 060 302 204 491 388 915 091 799 218 968 525 353 763 262 964 944 554 367 155 544 031 245 110 816 251 492 074 177 273 216 959 439 947 395 961 962 571 165 502 188 943 427 598 035 406 815 711 759 980 047 907 241 628 798 510 899 229 717 243 576 567 531 265 884 740 479 784 185 754 356 370 791 972 417 353 779 618 045 948 630 852 179 367 446 358 892 679 096 826 811 496 498 647 986 673 650 615 610 115 347 023 616 070 385 818 767 210 541 600 829 445 336 036 214 968 673 813 496 152 965 299 563 725 570 892 906 260 162 271 920 732 274 413 026 421 514 778 718 978 024 853 032 868 230 039 464 262 669 944 313 832 350 242 556 386 282 677 562

Out[◦]= {0, 0}

Out[◦]= 1 210 594 593 397 286 870 492 273 714 461 297 133 849 103 171 031 660 848 941 357 995 877 355 898 521 231 769 552 005 938 743 132 618 945 203 946 201 134 534 854 579 372 621 027 957 381 720 485 619 246 980 526 001 983 479 455 097 852 694 445 046 942 963 271 937 618 829 769 216 134 244 194 555 528 962 321 042 205 021 252 349 935 294 782 912 589 920 333 116 292 075 215 528 744 223 772 229 626 236 922 093 750 608 817 736 473 450 395 521 300 363 554 442 609 388 542 652 504 559 585 638 593 468 359 347 312 730 133 164 322 331 949 488 067 082 112 962 968 297 359 564 504 461 325 492 805 876 654 239 483

In[◦]:=
```
(* converting quad rels to rr format *)
MonsCD2 = {};
For[i = 1, i ≤ 19, i++, For[j = 1, j ≤ i, j++,
   AppendTo[MonsCD2, VarsCDleft[[i]] × VarsCDleft[[j]]];
  ]];
QuadEqswithrr = Sum[
   (Re[Coefficient[QuadEqs, MonsCD2[[k]]]] +
      rr Im[Coefficient[QuadEqs, MonsCD2[[k]]] / 2 ^ (1 / 2)])
    MonsCD2[[k]], {k, 1, Length[MonsCD2]}] ;
DeleteDuplicates[Simplify[QuadEqs - (QuadEqswithrr /. rr → (-2) ^ (1 / 2))]]
```

Out[◦]= {0}

```
In[•]:= QuadEqsMod19 = Sum[
          Mod[Coefficient[QuadEqswithrr /. rr → 6, MonsCD2[[k]]], 19] MonsCD2[[k]],
          {k, 1, Length[MonsCD2]}] ;
        (* now writing the equations mod 19
         the file will have cubic and quadratic
         equations and two versions of linear equations *)
        EqsMod19deg23 = Join[Eqs58mod19.AllMonsCD, QuadEqsMod19];

        fname = "Desktop/FPP2020/Eqs58mod19v1";
        OpenWrite[fname];
        Write[fname,
          {VarsCDleft, EqsMod19deg23, {EqLin1mod19, EqLin2mod19, EqLin3mod19},
           {EqLin1mod19, EqLin4mod19, EqLin5mod19}}];
        Close[fname];

        (* the first version gave the solutions *)

In[•]:= (* the below point was found by Magma by adding
         linear cuts and checking Hilbert polynomial *)
        PtMod19 = Solve[{c1 + 2 * d6 == 0, c4 + 16 * d6 == 0, c7 + 18 * d6 == 0, c10 + 15 * d6 == 0,
             c11 + 16 * d6 == 0, c12 + 15 * d6 == 0, c13 + 8 * d6 == 0, c14 + 8 * d6 == 0, c15 + 10 * d6 == 0,
             c16 + 2 * d6 == 0, c17 + 8 * d6 == 0, c18 + 4 * d6 == 0, c19 + 8 * d6 == 0, d1 + d6 == 0,
             d2 + 2 * d6 == 0, d3 + 15 * d6 == 0, d4 + 16 * d6 == 0, d5 + 14 * d6 == 0, d1 == 1}][[1]]
        Mod[EqsMod19deg23 /. PtMod19, 19]
        Mod[{EqLin1mod19, EqLin2mod19, EqLin3mod19} /. PtMod19, 19]

Out[•]= {c1 → 2, c10 → 15, c11 → 16, c12 → 15, c13 → 8, c14 → 8, c15 → 10, c16 → 2, c17 → 8,
         c18 → 4, c19 → 8, c4 → 16, c7 → 18, d1 → 1, d2 → 2, d3 → 15, d4 → 16, d5 → 14, d6 → -1}

Out[•]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Out[•]= {0, 0, 0}
```

```
In[◦]:= fname = "Desktop/FPP2020/UnivF";
OpenWrite[fname];
Write[fname, UnivF];
Close[fname];
fname = "Desktop/FPP2020/UnivD";
OpenWrite[fname];
Write[fname, UnivD];
Close[fname];

fname = "Desktop/FPP2020/EqsCD";
OpenWrite[fname];
Write[fname, Join[The58Eqswithrr /. rr → (-2) ^ (1 / 2),
   Join[QuadEqs, {EqLin1, EqLin2, EqLin3}]]];
Close[fname];
```

Now we will get to the business of building the actual triple cover. We first solve for f and d.

```
In[◦]:= fname = "Desktop/FPP2020/UnivF";
OpenRead[fname];
UnivF = Read[fname];
Close[fname];

fname = "Desktop/FPP2020/UnivD";
OpenRead[fname];
UnivD = Read[fname];
Close[fname];

fname = "Desktop/FPP2020/EqsCD";
OpenRead[fname];
EqsCD = Read[fname];
Close[fname];
```

```
In[•]:= VarsCDleft =
          {c1, c4, c7, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19, d1, d2, d3, d4, d5, d6};
        Length[VarsCDleft]
        Point19 =
         Solve[{c1 + 2 * d6 == 0, c4 + 16 * d6 == 0, c7 + 18 * d6 == 0, c10 + 15 * d6 == 0, c11 + 16 * d6 == 0,
            c12 + 15 * d6 == 0, c13 + 8 * d6 == 0, c14 + 8 * d6 == 0, c15 + 10 * d6 == 0, c16 + 2 * d6 == 0,
            c17 + 8 * d6 == 0, c18 + 4 * d6 == 0, c19 + 8 * d6 == 0, d1 + d6 == 0, d2 + 2 * d6 == 0,
            d3 + 15 * d6 == 0, d4 + 16 * d6 == 0, d5 + 14 * d6 == 0, d1 == 1}, Modulus → 19][[1]]
        AllMons3 = {};
        For[i = 1, i ≤ 13, i++, For[j = 1, j ≤ i, j++, For[k = 1, k ≤ j, k++,
            AppendTo[AllMons3, VarsCDleft[[i]] × VarsCDleft[[j]] × VarsCDleft[[k]]];
           ]]];
        For[i = 14, i ≤ 19, i++, For[j = 14, j ≤ i, j++, For[k = 14, k ≤ j, k++,
            AppendTo[AllMons3, VarsCDleft[[i]] × VarsCDleft[[j]] × VarsCDleft[[k]]];
           ]]];
        AllMons2 = {};
        For[i = 1, i ≤ 19, i++, For[j = 1, j ≤ i, j++,
           AppendTo[AllMons2, VarsCDleft[[i]] × VarsCDleft[[j]]];
          ]];
        AllMons1 = {};
        For[i = 1, i ≤ 19, i++,
          AppendTo[AllMons1, VarsCDleft[[i]]];
         ];
        EqsCDwithrr = {};
        Length[EqsCD];

        For[numeq = 1, numeq ≤ 73, numeq++,
          If[Mod[numeq, 20] == 10, Print[numeq]];
          AllMons = AllMons1;
          If[numeq ≤ 58, AllMons = AllMons3, If[numeq ≤ 70, AllMons = AllMons2]];
          eq = Expand[EqsCD[[numeq]]];
          eqwithrr = 0;
          For[k = 1, k ≤ Length[AllMons], k++,
           coef = Coefficient[eq, AllMons[[k]]] /. Table[VarsCDleft[[m]] → 0, {m, 1, 19}];
           eqwithrr += (Re[coef] + rr Im[coef / 2^(1 / 2)]) AllMons[[k]];
          ];
          AppendTo[EqsCDwithrr, eqwithrr];
         ];

        Simplify[(EqsCDwithrr /. rr → (-2)^(1 / 2)) - EqsCD]
Out[•]= 19
```

```
Out[•]= {c1 → 2, c10 → 15, c11 → 16, c12 → 15, c13 → 8, c14 → 8, c15 → 10, c16 → 2, c17 → 8,
  c18 → 4, c19 → 8, c4 → 16, c7 → 18, d1 → 1, d2 → 2, d3 → 15, d4 → 16, d5 → 14, d6 → 18}

10

30

50

70

Out[•]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

In[•]:= (* finding linearly independent equations near a point *)
GoodEqs = {};
runningmatrix = {};
rk = 0;
For[j = 73, j ≥ 1, j--,
  If[MatrixRank[Append[runningmatrix,
      Mod[Table[D[EqsCDwithrr[[j]] /. rr → 6, VarsCDleft[[k]]] /. Point19,
        {k, 1, 19}], 19]], Modulus → 19] > rk,
   AppendTo[runningmatrix, Mod[Table[D[EqsCDwithrr[[j]] /. rr → 6,
        VarsCDleft[[k]]] /. Point19, {k, 1, 19}], 19]];
   rk++;
   AppendTo[GoodEqs, EqsCDwithrr[[j]]];
   Print[{j, rk}];
  ]
 ];
(* finding (-2)^(1/2) mod powers of 19 using Newton's method *)
rvalue = 6;
For[i = 1, i ≤ 16, i++,
  rvalue += -(rvalue^2 + 2) / (2 rvalue);
 ];
acc = IntegerExponent[Numerator[Factor[rvalue^2 + 2]], 19]
rvalue =
  x /. Solve[Denominator[rvalue] x == Numerator[rvalue], Modulus → 19^acc][[1]];
IntegerExponent[rvalue^2 + 2, 19]
N[rvalue]
N[19^acc]

(* finding solutions modulo 19^20K using Newton's method *)
NewPt =
  {v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16, v17, v18, v19};
Pt = VarsCDleft /. Point19;
acc = 20 000;
GoodEqs1 = Join[{d1 - 1}, GoodEqs /. rr → rvalue];
```

```
For[i = 1, i ≤ 15, i++,
  Print[i];
  Defect = Mod[GoodEqs1 /. Table[VarsCDleft[[k]] → Pt[[k]], {k, 1, 19}], 19^acc];
  JacMat = Table[Mod[D[GoodEqs1[[n]], VarsCDleft[[m]]] /.
      Table[VarsCDleft[[k]] → Pt[[k]], {k, 1, 19}], 19^acc], {m, 1, 19}, {n, 1, 19}];
  Pt += NewPt /. (Solve[Table[0 == (Transpose[JacMat].NewPt + Defect)[[k]],
        {k, 1, 19}], Modulus → 19^acc][[1]]);
 ];

(* checking the answers *)
IntegerExponent[GoodEqs1 /. Table[VarsCDleft[[k]] → Pt[[k]], {k, 1, 19}], 19]

(* we will now try to translate from approximate p-adic to complex *)
smallsols = {};
rootscx = {};
For[i = 1, i ≤ 19, i++,
  FourNumbers = {Pt[[i]], 1, Mod[rvalue , 19^(acc - 1)], 19^(acc - 1)};
  vec1 = {f1, f2, f3, f4} /.
    Solve[{f1 == 1, f3 == 0, f4 == 0, {f1, f2, f3, f4}.FourNumbers == 0}][[1]];
  vec2 = {f1, f2, f3, f4} /. Solve[{f1 == 0, f3 == 1, f4 == 0,
       {f1, f2, f3, f4}.FourNumbers == 0}][[1]];
  vec3 = {f1, f2, f3, f4} /. Solve[{f1 == 0, f3 == 0, f4 == 1,
       {f1, f2, f3, f4}.FourNumbers == 0}][[1]];
  AppendTo[smallsols, LatticeReduce[{vec1, vec2, vec3}][[1]]];
  (* finding a small linear combination *)
  AppendTo[rootscx, -(smallsols[[i]][[2]] + Sqrt[-2] smallsols[[i]][[3]]) /
    (smallsols[[i]][[1]])];
 ];

Table[N[Denominator[rootscx[[k]]]], {k, 1, 19}]
RunningLCM = 1;
For[i = 1, i ≤ 19, i++, RunningLCM = LCM[RunningLCM, Denominator[rootscx[[i]]]]];
N[RunningLCM]
```

```
{73, 1}

{72, 2}

{71, 3}

{70, 4}

{69, 5}

{67, 6}

{64, 7}

{63, 8}
```

{61, 9}

{58, 10}

{57, 11}

{56, 12}

{55, 13}

{54, 14}

{53, 15}

{52, 16}

{51, 17}

{50, 18}

Out[∘]= 65 536

Out[∘]= 65 536

Out[∘]= $7.091457550406689 \times 10^{83\,803}$

Out[∘]= $2.488811727841071 \times 10^{83\,804}$

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

Out[∘]= {∞, 20 000, 20 000, 20 000, 20 000, 20 000, 20 000, 20 000, 20 000,
20 000, 20 000, 20 000, 20 000, 20 000, 20 000, 20 000, 20 000, 20 000, 20 000}

```
Out[•]= {7.665003340945076 × 10^869, 7.665003340945076 × 10^869, 3.740850825253819 × 10^866,
  7.665003340945076 × 10^869, 9.42800804827484 × 10^874, 8.485207243447358 × 10^875,
  8.485207243447358 × 10^875, 9.42800804827484 × 10^874, 2.828402414482453 × 10^875,
  9.42800804827484 × 10^874, 2.828402414482453 × 10^875, 3.142669349424947 × 10^874,
  2.063021454764736 × 10^872, 1., 7.254868213198218 × 10^441, 7.254868213198218 × 10^441,
  7.254868213198218 × 10^441, 5.560051195044769 × 10^447, 5.054591995495244 × 10^446}

Out[•]= 2.800118390337628 × 10^877

In[•]:= N[rootscx, 6]

Block[{$MaxExtraPrecision = 60 000},
 Expand[EqsCD /. Table[VarsCDleft[[l]] → rootscx[[l]], {l, 1, 19}]]]

Out[•]= {0.86326 - 1.81815 i, -0.000585772 - 0.000043303 i, 0.0389394 + 0.0135458 i,
  -0.000908508 - 0.000647927 i, 0.000086949 - 0.000690292 i, -0.00129332 + 0.00069011 i,
  -0.000028143 + 0.000161249 i, -0.001313310 + 0.000392808 i,
  -0.000382104 - 0.000848050 i, 0.00043571 + 0.00218287 i, 0.000497563 - 0.000280283 i,
  -0.000326105 + 0.000141537 i, -0.00172828 - 0.00098071 i, 1.00000,
  0.000309943 - 0.000015828 i, -0.0369797 - 0.0078264 i, 0.000696263 + 0.000095915 i,
  -0.000208183 + 0.000230775 i, 6.6071 × 10^-6 - 0.0000811679 i}

Out[•]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

In[•]:= (* we will not look for a nicer formula for f
  and will just proceed with the plan *)
VarsW = {W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11, W12, W13, W14};
AllMons6 = {};
For[i = 1, i ≤ 4, i++, For[j = 1, j ≤ i, j++, For[k = 1, k ≤ j, k++,
    AppendTo[AllMons6, VarsW[[i]] × VarsW[[j]] × VarsW[[k]]];
   ]]];
For[i = 5, i ≤ 14, i++, For[j = 5, j ≤ i, j++,
   AppendTo[AllMons6, VarsW[[i]] × VarsW[[j]]];
  ]];

Sectionf = UnivF /. Table[VarsCDleft[[l]] → rootscx[[l]], {l, 1, 19}];
Sectiond = UnivD /. Table[VarsCDleft[[l]] → rootscx[[l]], {l, 1, 19}];
Numf = Sectionf (W2 + W3 + W4);
Numf = Numerator[Factor[Numf]];
N[Max[Abs[Coefficient[Numf, AllMons6]]]]

Out[•]= 1.683445534180258 × 10^955
```

We will find multiple points on C, sigma(C) and sigma^2(C).

```
In[∘]:= (* finding points on C1 *)
CyclicW = {W2 → W3, W3 → W4, W4 → W2, W5 → W6, W6 → W7,
    W7 → W5, W8 → W9, W9 → W10, W10 → W8, W11 → W12, W12 → W13, W13 → W11};

SetOfPointsC = {};
SetOfPointsSigmaC = {};
SetOfPointsSigma2C = {};
prec = 6000;

For[numpt = 1, numpt ≤ 100, numpt++,
  If[Mod[numpt, 20] == 0, Print[numpt]];
  rpt1 = (RandomInteger[100 000] + 1) / 10 000 + I (RandomInteger[100 000] + 1) / 10 000;
  RelsToSolve = Join[Table[EqsFPPC3W[[i]] == 0, {i, 1, 38}],
    {0 == Sectionf, 0 == Sectiond, W2 == rpt1, W2 + W3 + W4 == 1}];
  (* so that we don't pick up solutions of upexact with W2+W3+W4=0 *)
  sols = NSolve[RelsToSolve, 600][[1]];
  If[Abs[Sectionf /. sols] < 10^(-40), (* to avoid some spurrious results *)
   fr = FindRoot[Join[Table[EqsFPPC3W[[i]] == 0, {i, 1, 11}], {0 == Sectiond, W2 == rpt1,
        W2 + W3 + W4 == 1}], Table[{VarsW[[k]], (VarsW[[k]] /. sols)}, {k, 1, 14}],
      AccuracyGoal → prec, WorkingPrecision → (3 / 2) prec, PrecisionGoal → prec];
   (* it is better to use Sectiond which doesn't have the multiple root *)
   If[Sum[Abs[EqsFPPC3W[[k]] /. fr], {k, 1, 38}] ≤ 10^(-prec + 1),
    AppendTo[SetOfPointsC, N[fr, prec]];
    AppendTo[SetOfPointsSigma2C, N[fr, prec] /. CyclicW];
    AppendTo[SetOfPointsSigmaC, N[fr, prec] /. CyclicW /. CyclicW];
   ];
  ];
 ];
Print[Length[SetOfPointsC], " points"];
```

```
20
40
60
80
100
100 points
```

Defining sigma_lift action and the cube of z.

In[∘]:=
```
SubSigma = Append[CyclicW, z → z Sectiond / SectionSigmaf];
zcube = SectionSigmaf / Sectionf;
SectionSigmaf = Sectionf /. CyclicW;
SectionSigma2f = SectionSigmaf /. CyclicW;
N[(((z^3 /. SubSigma) /. z^3 -> SectionSigmaf / Sectionf) - (zcube /. SubSigma)) /.
  SetOfPointsWextra[[45]]]
```

Out[∘]=
```
0. + 0. i
```

Finding conditions of tangency at C, sigma(C), sigma^2(C) in the form of a tangent vector to the surface which is normal to the curve.

In[∘]:=
```
TangentsAtC = {};
For[i = 1, i ≤ 100, i++,
  If[Mod[i, 20] == 0, Print[i]];
  pt = (VarsW /. SetOfPointsC[[i]]) +
    t {0, 0, 1, dW4, dW5, dW6, dW7, dW8, dW9, dW10, dW11, dW12, dw13, dw14};
  pts = Table[VarsW[[k]] → pt[[k]], {k, 1, 14}];
  EqsPlus = Table[EqsFPPC3W[[i]], {i, 1, 11}];
  sols = Solve[
     Table[0 == SeriesCoefficient[EqsPlus[[j]] /. pts, {t, 0, 1}], {j, 1, 11}]][[1]];
  AppendTo[TangentsAtC, pts /. sols];
 ];
TangentsAtSigma2C = TangentsAtC /. CyclicW;
TangentsAtSigmaC = TangentsAtC /. CyclicW /. CyclicW;
```

```
20
40
60
80
100
```

In[∘]:=
```
(* sanity check *)
N[Series[Sectionf /. TangentsAtC[[10]], {t, 0, 2}]]
N[Series[SectionSigma2f /. TangentsAtSigmaC[[10]], {t, 0, 2}]]
N[Series[SectionSigmaf /. TangentsAtSigma2C[[10]], {t, 0, 2}]]
```

Out[∘]= $(0.00406326 + 0.00653419\, i)\, (t + 0.)^2 + O[t + 0.]^3$

Out[∘]= $(0.00406326 + 0.00653419\, i)\, (t + 0.)^2 + O[t + 0.]^3$

Out[∘]= $(0.00406326 + 0.00653419\, i)\, (t + 0.)^2 + O[t + 0.]^3$

In[∘]:=
```
AllMons =
  MonomialList[(W1 + W2 + W3 + W4) ^3 + (W5 + W6 + W7 + W8) (W9 + W10 + W11 + W12 + W13 + W14)];
maxlength = 500;
CoeffList = Table[ToExpression[StringJoin["A", ToString[i]]], {i, 1, maxlength}];
(* experimentally found to give one solution up to scaling *)
count = Length[AllMons]
AllMons = Table[
   AllMons[[i]] / (AllMons[[i]] /. Table[VarsW[[k]] → 1, {k, 1, 14}]), {i, 1, count}];

UnivPoly = Sum[CoeffList[[i]] × AllMons[[i]], {i, 1, Length[AllMons]}];
Eqs = Join[Table[0 == UnivPoly /. SetOfPointsC[[i]], {i, 1, 81}],
   Table[0 == UnivPoly /. SetOfPointsSigmaC[[i]], {i, 1, 81}]];
Eqs = Join[Eqs, Table[0 == SeriesCoefficient[UnivPoly /. TangentsAtC[[i]], {t, 0, 1}],
    {i, 1, 100}]];
sols = Chop[Solve[Eqs]][[1]];
UnivPolySolved = UnivPoly /. sols;

PossibleEqs = {};
For[j = 1, j ≤ Length[AllMons], j ++,
  UPs = Coefficient[UnivPolySolved, CoeffList[[j]]];
  If[Abs[UPs /. SetOfPointsWextra[[1]]] > .1,
   Print[j];
   Block[{$MaxExtraPrecision = 10 000},
    AppendTo[PossibleEqs, Sum[RootApproximant[Coefficient[UPs, AllMons[[i]]], 2] ×
       AllMons[[i]], {i, 1, Length[AllMons]}]]
   ];
  ];
 ];
U4 = Factor[z PossibleEqs[[1]] / Sectiond];
U5 = Factor[U4 /. SubSigma];
U6 = Factor[U5 /. SubSigma];
```

Out[∘]= 44

1

In[•]:=
```
AllMons =
  MonomialList[(W1 + W2 + W3 + W4) ^3 + (W5 + W6 + W7 + W8) (W9 + W10 + W11 + W12 + W13 + W14)];
(* experimentally found to give one solution up to scaling *)
count = Length[AllMons]
AllMons = Table[
   AllMons[[i]] / (AllMons[[i]] /. Table[VarsW[[k]] → 1, {k, 1, 14}]), {i, 1, count}];

UnivPoly = Sum[CoeffList[[i]] × AllMons[[i]], {i, 1, Length[AllMons]}];
Eqs = Join[Table[0 == UnivPoly /. SetOfPointsSigma2C[[i]], {i, 1, 81}],
   Table[0 == UnivPoly /. SetOfPointsSigmaC[[i]], {i, 1, 100}]];
Eqs = Join[Eqs, Table[0 == SeriesCoefficient[
      UnivPoly /. TangentsAtSigma2C[[i]], {t, 0, 1}], {i, 1, 81}]];
sols = Chop[Solve[Eqs]][[1]];
UnivPolySolved = UnivPoly /. sols;

PossibleEqs = {};
For[j = 1, j ≤ Length[AllMons], j++,
  UPs = Coefficient[UnivPolySolved, CoeffList[[j]]];
  If[Abs[UPs /. SetOfPointsWextra[[45]]] > .1,
   Print[j];
   Block[{$MaxExtraPrecision = 10 000},
    AppendTo[PossibleEqs, Sum[RootApproximant[Coefficient[UPs, AllMons[[i]]], 2] ×
       AllMons[[i]], {i, 1, Length[AllMons]}]]
   ];
  ];
 ];
U7 = Factor[z^(-1) PossibleEqs[[1]] / Sectiond] ;
U8 = Factor[ U7 /. SubSigma];
U9 = Factor[U8 /. SubSigma];
```

Out[•]= 44

1

Now we find the cubic equations. We will call the variables P0,...,P9. This calculation takes a long time, be patient.

In[•]:=
```
(* make points on FPP *)
prec = Round[Accuracy[SetOfPointsWextra]] - 1
SetOfPointsW = N[SetOfPointsWextra, prec];
SetOfPointsZ = Table[
   Append[SetOfPointsW[[i]], z -> N[(zcube /. SetOfPointsW[[i]])^(1 / 3), prec]],
```

```
    {i, 1, Length[SetOfPointsW]}];
SetOfPointsP = Table[{P0 → W1 /. SetOfPointsZ[[i]], P1 → W2 /. SetOfPointsZ[[i]],
     P2 → W3 /. SetOfPointsZ[[i]], P3 → W4 /. SetOfPointsZ[[i]],
     P4 → U4 /. SetOfPointsZ[[i]], P5 → U5 /. SetOfPointsZ[[i]], P6 → U6 /.
      SetOfPointsZ[[i]], P7 → U7 /. SetOfPointsZ[[i]], P8 → U8 /. SetOfPointsZ[[i]],
     P9 → U9 /. SetOfPointsZ[[i]]}, {i, 1, Length[SetOfPointsZ]}];

maxlength = 500;
CoeffList = Table[ToExpression[StringJoin["A", ToString[i]]], {i, 1, maxlength}];

Beautify[x_] :=
  Ret = Expand[Rationalize[Chop[N[Re[x], Round[prec / 2]]]] +
     Rationalize[Chop[N[Im[x] / √2, Round[prec / 2]]]] I √2];

deg = 3;
VarsP = {P0, P1, P2, P3, P4, P5, P6, P7, P8, P9};
AllMons = MonomialList[(P0 + P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9) ^deg];
count = Length[AllMons]
AllMons = Table[
   AllMons[[i]] / (AllMons[[i]] /. Table[VarsP[[m]] → 1, {m, 1, 10}]), {i, 1, count}];
WtinZ = {0, 0, 0, 0, 1, 1, 1, -1, -1, -1};

EqsFPP = {};
For[k = 0, k ≤ 2, k++,
  UnivPolyP = 0;
  Print[k];
  count1 = 0;
  For[j = 1, j ≤ count, j++,
   expall = Sum[Exponent[AllMons[[j]], VarsP[[m]]] × WtinZ[[m]], {m, 1, 10}];
   If[Mod[expall, 3] == k,
    UnivPolyP += CoeffList[[j]] × AllMons[[j]];
    count1++;
   ];
  ];

  Eqs = Table[0 == UnivPolyP /. SetOfPointsP[[l]], {l, 1, count1 + 10}];
  sols = Block[{$MaxExtraPrecision = 10000}, NSolve[Eqs][[1]]];
  UpSolved = UnivPolyP /. sols;
  For[j = 1, j ≤ count, j++,
   neweq = Coefficient[UpSolved, CoeffList[[j]]];
   If[ByteCount[neweq] > 100,
    Block[{$MaxExtraPrecision = 10000}, AppendTo[EqsFPP, Sum[
        Beautify[Coefficient[neweq, AllMons[[l]]]] × AllMons[[l]], {l, 1, count}]]
```

```
    ];
   ];
  ];

 ];

Print[Length[EqsFPP]];
Table[Accuracy[EqsFPP[[k]]], {k, 1, 84}]
Table[ByteCount[EqsFPP[[k]]], {k, 1, 84}]
EqsFPP = Table[Numerator[Factor[EqsFPP[[k]]]], {k, 1, 84}];
Table[ByteCount[EqsFPP[[k]]], {k, 1, 84}]
```

```
Out[∘]= 59 995

Out[∘]= 220

0

1

2

84

Out[∘]= {∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞,
 ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞,
 ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞}

Out[∘]= {327 032, 325 496, 325 496, 302 648, 302 648, 302 648, 302 648, 302 648, 302 648, 302 648,
 302 648, 302 648, 348 072, 348 032, 326 264, 326 712, 325 752, 305 832, 322 216, 322 216,
 305 832, 322 216, 322 216, 305 832, 322 216, 322 216, 358 976, 359 032, 359 032, 358 976,
 235 120, 235 120, 235 152, 235 056, 235 056, 235 056, 235 056, 235 024, 238 864,
 238 992, 238 992, 239 248, 239 184, 238 928, 305 168, 304 912, 235 056, 235 056,
 235 024, 235 088, 235 088, 235 056, 286 112, 286 048, 263 168, 286 112, 263 168,
 235 184, 235 200, 235 152, 235 120, 235 088, 235 104, 235 136, 235 120, 240 208,
 239 376, 238 864, 239 696, 239 376, 239 568, 234 800, 235 152, 235 136, 235 120,
 235 152, 235 136, 235 120, 291 376, 291 312, 291 312, 291 376, 291 312, 291 376}

Out[∘]= {281 216, 281 216, 281 216, 302 912, 302 912, 302 912, 302 912, 302 912, 302 912, 302 912,
 302 912, 302 912, 310 160, 310 160, 281 216, 281 152, 281 216, 310 096, 332 368, 332 368,
 310 096, 332 368, 332 368, 310 096, 332 368, 332 368, 324 728, 324 736, 324 736, 324 728,
 221 280, 221 280, 221 280, 221 216, 221 216, 221 216, 221 216, 221 216, 221 280,
 221 216, 221 216, 221 280, 221 216, 221 280, 254 848, 254 848, 221 216, 221 216,
 221 216, 221 216, 221 216, 221 216, 249 200, 249 136, 249 136, 249 200, 249 136,
 221 408, 221 408, 221 344, 221 344, 221 344, 221 344, 221 344, 221 344, 221 408,
 221 344, 221 344, 221 408, 221 344, 221 408, 208 384, 221 344, 221 344, 221 344,
 221 344, 221 344, 221 344, 254 976, 254 912, 254 912, 254 976, 254 912, 254 976}
```

In[◦]:=
```
MatrixRank[Table[D[EqsFPP, VarsP[[k]]] /. {P0 → 0, P1 → 1, P2 → 0,
    P3 → 0, P4 → 0, P5 → 0, P6 → 0, P7 → 0, P8 → 0, P9 → 0}, {k, 1, 10}]]
ns = NullSpace[Transpose[Table[D[EqsFPP, VarsP[[k]]] /. {P0 → 0, P1 → 1, P2 → 0,
       P3 → 0, P4 → 0, P5 → 0, P6 → 0, P7 → 0, P8 → 0, P9 → 0}, {k, 1, 10}]]];
ns = RootReduce[
   ns];
```

Out[◦]= 7

In[◦]:=
```
w = Exp[2 Pi I / 3]
slsfixedapprox1 = NSolve[Table[0 == EqsFPP[[k]] /.
      {P0 → 1, P2 → P1, P3 → P1, P5 → P4, P6 → P4, P8 → P7, P9 → P7}, {k, 1, 84}], 20 000];
slsfixed1 = VarsP /. {P0 → 1, P2 → P1, P3 → P1, P5 → P4, P6 → P4, P8 → P7, P9 → P7} /.
   slsfixedapprox1;
slsfixedapprox2 = NSolve[Table[0 == EqsFPP[[k]] /. {P0 → 1, P2 → P1, P3 → P1,
       P5 → w P4, P6 → w^2 P4, P8 → w^2 P7, P9 → w P7}, {k, 1, 84}], 20 000];
slsfixed2 = VarsP /. {P0 → 1, P2 → P1, P3 → P1, P5 → w P4, P6 → w^2 P4,
      P8 → w^2 P7, P9 → w P7} /. slsfixedapprox2;
slsfixedapprox3 = NSolve[Table[0 == EqsFPP[[k]] /. {P0 → 1, P2 → P1, P3 → P1,
       P5 → w^2 P4, P6 → w P4, P8 → w P7, P9 → w^2 P7}, {k, 1, 84}], 20 000];
slsfixed3 = VarsP /. {P0 → 1, P2 → P1, P3 → P1, P5 → w^2 P4, P6 → w P4,
      P8 → w P7, P9 → w^2 P7} /. slsfixedapprox3;
value1 = Beautify[ slsfixed1[[3]]];
value2 = Block[{$MaxExtraPrecision = 10 000},
   RootApproximant [w slsfixed2[[1]] + w^2 slsfixed3[[1]], 2]];
value3 = Block[{$MaxExtraPrecision = 10 000},
   RootApproximant [w^2 slsfixed2[[1]] + w slsfixed3[[1]], 2]];

help456 =
  {c1, c2, c3} /. Solve[{1 == {c1, c2, c3}.{value1[[5]], value1[[6]], value1[[7]]},
      1 == {c1, c2, c3}.{value2[[5]], value2[[6]], value2[[7]]},
      1 == {c1, c2, c3}.{value3[[5]], value3[[6]], value3[[7]]}}][[1]];
help789 =
  {c1, c2, c3} /. Solve[{1 == {c1, c2, c3}.{value1[[8]], value1[[9]], value1[[10]]},
      1 == {c1, c2, c3}.{value2[[8]], value2[[9]], value2[[10]]},
      1 == {c1, c2, c3}.{value3[[8]], value3[[9]], value3[[10]]}}][[1]];
```

Out[◦]= $e^{\frac{2 i \pi}{3}}$

Now we will recompute the equations but in better variables.

In[◦]:=
```
(* make points on FPP *)
prec = 1000;
SetOfPointsW = N[SetOfPointsWextra, prec];
SetOfPointsZ = Table[
```

```
    Append[SetOfPointsW[[i]], z -> N[(zcube /. SetOfPointsW[[i]]) ^ (1 / 3), prec]],
    {i, 1, Length[SetOfPointsW]}];
SetOfPointsQ = Table[{Q0 → W1 /. SetOfPointsZ[[i]], Q1 → W2 /. SetOfPointsZ[[i]],
     Q2 → W3 /. SetOfPointsZ[[i]], Q3 → W4 /. SetOfPointsZ[[i]],

     Q4 → help456.{ U4, U5, U6} /. SetOfPointsZ[[i]],
     Q5 → help456.{ U5, U6, U4} /. SetOfPointsZ[[i]],
     Q6 → help456.{ U6, U4, U5} /. SetOfPointsZ[[i]],

     Q7 → help789.{ U7, U8, U9} /. SetOfPointsZ[[i]],
     Q8 → help789.{ U8, U9, U7} /. SetOfPointsZ[[i]],
     Q9 → help789.{ U9, U7, U8} /. SetOfPointsZ[[i]]},

    {i, 1, Length[SetOfPointsZ]}];


maxlength = 500;
CoeffList = Table[ToExpression[StringJoin["A", ToString[i]]], {i, 1, maxlength}];

Beautify[x_] :=
  Ret = Expand[Rationalize[Chop[N[Re[x], Round[prec / 2]]]] +
      Rationalize[Chop[N[Im[x] / √2 , Round[prec / 2]]]] I √2 ];

deg = 3;
VarsQ = {Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9};
AllMons = MonomialList[Sum[VarsQ[[k]], {k, 1, 10}] ^ deg];
count = Length[AllMons]
AllMons = Table[
    AllMons[[i]] / (AllMons[[i]] /. Table[VarsQ[[m]] → 1, {m, 1, 10}]), {i, 1, count}];
WtinZ = {0, 0, 0, 0, 1, 1, 1, -1, -1, -1};

EqsFPPQ = {};
For[k = 0, k ≤ 2, k++,
  UnivPolyQ = 0;
  Print[k];
  count1 = 0;
  For[j = 1, j ≤ count, j++,
   expall = Sum[Exponent[AllMons[[j]], VarsQ[[m]]] × WtinZ[[m]], {m, 1, 10}];
   If[Mod[expall, 3] == k,
    UnivPolyQ += CoeffList[[j]] × AllMons[[j]];
    count1++;
   ];
  ];
```

```
  Eqs = Table[0 == UnivPolyQ /. SetOfPointsQ[[l]], {l, 1, count1 + 10}];
  sols = Block[{$MaxExtraPrecision = 10 000}, NSolve[Eqs][[1]]];
  UpSolved = UnivPolyQ /. sols;
  For[j = 1, j ≤ count, j++,
   neweq = Coefficient[UpSolved, CoeffList[[j]]];
   If[ByteCount[neweq] > 100,
    Block[{$MaxExtraPrecision = 10 000}, AppendTo[EqsFPPQ, Sum[
        Beautify[Coefficient[neweq, AllMons[[l]]]] × AllMons[[l]], {l, 1, count}]]
     ];
   ];
  ];

 ];

Print[Length[EqsFPPQ]];
Table[Accuracy[EqsFPPQ[[k]]], {k, 1, 84}]
Table[ByteCount[EqsFPPQ[[k]]], {k, 1, 84}]
EqsFPPQ = Table[Numerator[Factor[EqsFPPQ[[i]]]], {i, 1, 84}];
Table[ByteCount[EqsFPPQ[[k]]], {k, 1, 84}]
```

Out[∘]= 220

0

1

2

84

Out[∘]= {∞, ∞,
∞, ∞,
∞, ∞}

Out[∘]= {32 632, 32 760, 32 632, 32 608, 32 776, 32 608, 32 712, 32 560, 32 560, 32 680, 32 592,
32 672, 33 320, 33 064, 32 728, 32 728, 32 824, 33 176, 33 288, 33 224, 33 216, 33 240,
33 200, 33 280, 33 216, 33 176, 32 584, 32 656, 32 592, 32 472, 32 184, 32 176, 32 248,
32 128, 32 192, 32 240, 32 192, 32 176, 32 272, 32 304, 32 208, 32 352, 32 288, 32 320,
33 760, 33 864, 32 224, 32 272, 32 128, 32 240, 32 288, 32 272, 33 000, 32 832,
32 768, 32 960, 32 784, 31 992, 32 016, 32 240, 32 144, 32 208, 32 192, 32 256,
32 176, 32 328, 32 344, 32 344, 32 376, 32 336, 32 440, 33 344, 32 168, 32 304,
32 384, 32 264, 32 336, 32 224, 33 776, 34 000, 33 936, 34 112, 33 896, 34 088}

```
Out[•]= {23 496, 23 496, 23 496, 23 432, 23 432, 23 432, 23 432, 23 432, 23 432, 23 432, 23 432,
  23 432, 24 104, 24 104, 23 496, 23 432, 23 496, 24 040, 24 192, 24 152, 24 040, 24 192,
  24 280, 24 040, 24 256, 24 216, 23 488, 23 496, 23 496, 23 488, 23 616, 23 616, 23 616,
  23 552, 23 552, 23 552, 23 552, 23 552, 23 616, 23 552, 23 552, 23 616, 23 552, 23 616,
  25 528, 25 528, 24 080, 24 080, 24 088, 24 064, 24 008, 23 976, 24 752, 24 640,
  24 368, 24 752, 24 320, 23 600, 23 656, 23 680, 23 680, 23 680, 23 680, 23 680,
  23 680, 23 744, 23 680, 23 680, 23 744, 23 680, 23 744, 24 928, 23 680, 23 680,
  23 680, 23 680, 23 680, 23 680, 25 528, 25 456, 25 424, 25 624, 25 432, 25 544}
```

```
In[•]:= (* this is not as good we hoped for, but will work in a pinch *)
EqsFPPwithrr = {};
For[numeq = 1, numeq ≤ 84, numeq++,
  eq = Expand[EqsFPPQ[[numeq]]];
  eqwithrr = 0;
  For[k = 1, k ≤ Length[AllMons], k++,
   coef = Coefficient[eq, AllMons[[k]]];
   eqwithrr += (Re[coef]  + rr Im[coef / 2^(1 / 2)]) AllMons[[k]];
  ];
  AppendTo[EqsFPPwithrr, eqwithrr];
 ];
(* just checking *)
Simplify[(EqsFPPwithrr /. rr → (-2)^(1 / 2)) - EqsFPPQ]
```

```
Out[•]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

```
In[•]:= fname = "Desktop/FPP2020/EqsFPPwithrr";
OpenWrite[fname];
Write[fname, EqsFPPwithrr];
Close[fname];
```