Computational Companion to "Quasi-Symmetric Nets: A Constructive Approach to the Equimodular Elliptic Type of Kokotsakis Polyhedra" — Example 2 Helper

A. Nurmatov, M. Skopenkov, F. Rist, J. Klein, D. L. Michels

Tested on: Mathematica 14.0

In[774]:=

```
(*============================================================================
 =====*)
(*============================================================================
 =====*)
(*============================================================================
 =====*)
(*Quit*)
(*All angle sets in degrees*)
alpha1 = 15;
beta1 = 60;
gamma1 = 75;
delta1 = 90;

sigma1 = (alpha1 + beta1 + gamma1 + delta1) / 2;

anglesDeg = {
   {alpha1, beta1, gamma1, delta1}, (*Vertex 1*)
   {delta1, gamma1, beta1, alpha1}, (*Vertex 2*)
   {180 - delta1, 180 - gamma1, 180 - beta1, 180 - alpha1}, (*Vertex 3*)
   {alpha1, beta1, gamma1, delta1} (*Vertex 4*)};

(*------------------------------------------*)
```

```
(*Function to compute sigma from 4 angles*)
computeSigma[{α_, β_, γ_, δ_}] := (α + β + γ + δ) / 2;

(*------------------------------------------*)
(*Function to compute a,b,c,d from angles*)
computeABCD[{α_, β_, γ_, δ_}] :=
  Module[{alpha = α Degree, beta = β Degree, gamma = γ Degree,
    delta = δ Degree, sigma}, sigma = computeSigma[{α, β, γ, δ} Degree];
   {Sin[alpha] / Sin[sigma - alpha], Sin[beta] / Sin[sigma - beta],
    Sin[gamma] / Sin[sigma - gamma], Sin[delta] / Sin[sigma - delta]}];

(*------------------------------------------*)
(*Compute all σ values*)
sigmas = computeSigma /@ anglesDeg;

(*Compute all {a,b,c,d} values*)
results = computeABCD /@ anglesDeg;

(*Optional:extract individual values*)
{a1, b1, c1, d1} = results[[1]];
{a2, b2, c2, d2} = results[[2]];
{a3, b3, c3, d3} = results[[3]];
{a4, b4, c4, d4} = results[[4]];
{σ1, σ2, σ3, σ4} = sigmas;


(*========================================================================
 =====*)
(*================================
   CONDITION (N.0) ================================*)
(*========================================================================
 =====*)
uniqueCombos = {{1, 1, 1, 1}, {1, 1, 1, -1}, {1, 1, -1, -1}, {1, 1, -1, 1},
   {1, -1, 1, 1}, {1, -1, -1, 1}, {1, -1, 1, -1}, {1, -1, -1, -1}};

checkConditionN0Degrees[{α_, β_, γ_, δ_}] :=
  Module[{angles = {α, β, γ, δ}, results},
   results = Mod[uniqueCombos.angles, 360] // Chop;
   ! MemberQ[results, 0]];

conditionsN0 = checkConditionN0Degrees /@ anglesDeg;
allVerticesPass = And @@ conditionsN0;

Column[{TextCell[Style["================= CONDITION (N.0) =================",
    Darker[Green], Bold, 16], "Text"],
  If[allVerticesPass,
   Style["✓ All vertices satisfy (N.0).", Darker[Green], Bold],
```

```
   Style["✘ Some vertices fail (N.0).", Red, Bold]]}]


(*==========================================================================
 =====*)
(*===============================
   CONDITION (N.3)================================*)
(*==========================================================================
 =====*)
Ms = FullSimplify[Times @@@ results];
allEqualQ = Simplify[Equal @@ Ms];

Column[{TextCell[Style["================= CONDITION (N.3) =================",
     Blue, Bold, 16], "Text"], If[allEqualQ,
   Row[{Style["✔ M1 = M2 = M3 = M4 = ", Bold], Ms[[1]]}],
   Style["✘ M_i are not all equal.", Red, Bold]]}]


(*==========================================================================
 =====*)
(*===========================CONDITION (N.4)===========================*)
(*==========================================================================
 =====*)
aList = results[[All, 1]];
cList = results[[All, 3]];
dList = results[[All, 4]];

rList = FullSimplify /@ (aList * dList); {r1, r2, r3, r4} = rList;
sList = FullSimplify /@ (cList * dList); {s1, s2, s3, s4} = sList;

Column[{TextCell[Style["================= CONDITION (N.4) =================",
     Darker[Blue], Bold, 16], "Text"],
  If[r1 == r2 && r3 == r4 && s1 == s4 && s2 == s3, Column[
    {Row[{Style["✔ r1 = r2 = ", Bold], r1, Style["; ✔ r3 = r4 = ", Bold], r3}],
     Row[{Style["✔ s1 = s4 = ", Bold], s1, Style["; ✔ s2 = s3 = ", Bold],
       s2}]}], Style["✘ Condition (N.4) fails.", Red, Bold]]
 }]


(*==========================================================================
 =====*)
(*=================================
  CONDITION (N.5)==============================*)
(*==========================================================================
 =====*)
fList = FullSimplify /@ (aList * cList); {f1, f2, f3, f4} = fList;
M1 = Ms[[1]];
```

```
m[M1_] := Piecewise[{{1 - M1, M1 < 1}, {(M1 - 1) / M1, M1 > 1}}] // N;
K = EllipticK[m[M1]] // N;
Kp = EllipticK[1 - m[M1]] // N;

computeTi[i_] :=
  Module[{sigma = sigmas⟦i⟧, r = rList⟦i⟧, s = sList⟦i⟧, f = fList⟦i⟧, base},
   base = I * Im[InverseJacobiDN[Piecewise[
         {{Sqrt[f], M1 < 1}, {1 / Sqrt[f], M1 > 1}}], m[M1]]] / Kp;
   Which[sigma < 180, Which[r > 1 && s > 1, base, (r < 1 && s > 1) || (r > 1 && s < 1),
     1 + base, r < 1 && s < 1, 2 + base], sigma > 180, Which[r > 1 && s > 1,
     2 + base, (r < 1 && s > 1) || (r > 1 && s < 1), 3 + base, r < 1 && s < 1, base]]];

tList = computeTi /@ Range[4];
RoundWithTolerance[x_, tol_ : 10^-6] := Module[{nearest}, nearest = Round[x];
   If[Abs[x - nearest] ≤ tol, nearest, x]];

checkValidCombination[M1_, ε_ : 10^-6] :=
  Module[{i, combo, dotProd, rePart, imPart, n2, expr, foundQ = False},
   Print[Style["⚠ Approximate validation using ε-tolerance. For rigorous
       proof, see the referenced paper.", Darker@Orange, Italic]];
   Do[combo = uniqueCombos⟦i⟧;
    dotProd = tList.combo;
    rePart = Abs[Re[dotProd]];
    imPart = Abs[Im[dotProd]];
    If[M1 < 1,
     If[Mod[RoundWithTolerance[rePart], 4] < ε,
      If[Mod[RoundWithTolerance[imPart], 2] < ε, expr =
        tList⟦1⟧ + combo⟦2⟧ × tList⟦2⟧ + combo⟦3⟧ × tList⟦3⟧ + combo⟦4⟧ × tList⟦4⟧;
       Print[Style["✅ Valid Combination Found (M < 1):",
          Darker[Green], Bold], "\n", Style["e1 = ", Bold], combo⟦2⟧,
        Style[", e2 = ", Bold], combo⟦3⟧, Style[", e3 = ", Bold], combo⟦4⟧,
        "\n", Style["t1 = ", Bold], Re[tList⟦1⟧], "K + ", Im[tList⟦1⟧], "iK'",
        "\n", Style["t2 = ", Bold], Re[tList⟦2⟧], "K + ", Im[tList⟦2⟧], "iK'",
        "\n", Style["t3 = ", Bold], Re[tList⟦3⟧], "K + ", Im[tList⟦3⟧], "iK'",
        "\n", Style["t4 = ", Bold], Re[tList⟦4⟧], "K + ", Im[tList⟦4⟧],
        "iK'", "\n", Style["t1 + e1*t2 + e2*t3 + e3*t4 = ", Bold],
        Re[expr], "K + ", Im[expr], "iK'"];
       foundQ = True;
       Break[]]]];
    If[M1 > 1,
     If[Mod[RoundWithTolerance[imPart], 2] < ε,
      n2 = Quotient[RoundWithTolerance[imPart], 2];
      If[Mod[RoundWithTolerance[Abs[rePart - 2 n2]], 4] < ε, expr =
        tList⟦1⟧ + combo⟦2⟧ × tList⟦2⟧ + combo⟦3⟧ × tList⟦3⟧ + combo⟦4⟧ × tList⟦4⟧;
```

```
        Print[Style["✅ Valid Combination Found (M > 1):",
          Darker[Green], Bold], "\n", Style["e1 = ", Bold], combo[[2]],
         Style[", e2 = ", Bold], combo[[3]], Style[", e3 = ", Bold], combo[[4]],
         "\n", Style["t1 = ", Bold], Re[tList[[1]]], "K + ", Im[tList[[1]]], "iK'",
         "\n", Style["t2 = ", Bold], Re[tList[[2]]], "K + ", Im[tList[[2]]], "iK'",
         "\n", Style["t3 = ", Bold], Re[tList[[3]]], "K + ", Im[tList[[3]]], "iK'",
         "\n", Style["t4 = ", Bold], Re[tList[[4]]], "K + ", Im[tList[[4]]],
         "iK'", "\n", Style["t1 + e1*t2 + e2*t3 + e3*t4 = ", Bold],
         Re[expr], "K + ", Im[expr], "iK'"];
        foundQ = True;
        Break[]]]], {i, Length[uniqueCombos]}];
    If[! foundQ, Print[Style["❌ No valid combination found.", Red, Bold], "\n",
      Style["t1 = ", Bold], Re[tList[[1]]], "K + ", Im[tList[[1]]], "iK'", "\n",
      Style["t2 = ", Bold], Re[tList[[2]]], "K + ", Im[tList[[2]]], "iK'", "\n",
      Style["t3 = ", Bold], Re[tList[[3]]], "K + ", Im[tList[[3]]], "iK'", "\n",
      Style["t4 = ", Bold], Re[tList[[4]]], "K + ", Im[tList[[4]]], "iK'", "\n"]];];


Column[{TextCell[Style["================= CONDITION (N.5) =================",
     Purple, Bold, 16], "Text"]}]
res = checkValidCombination[M1];


(*============================================================================
 =====*)
(*================================
    OTHER PARAMETER=================================*)
(*============================================================================
 =====*)
Column[
 {TextCell[Style["================= OTHER PARAMETERS =================",
     Darker[Purple], Bold, 16], "Text"],
  Row[{Style["u = ", Bold], 1 - M1}],
  Row[{Style["σ1 = ", Bold], σ1 Degree, Style[", σ2 = ", Bold], σ2 Degree,
     Style[", σ3 = ", Bold], σ3 Degree, Style[", σ4 = ", Bold], σ4 Degree}],
  Row[{Style["f1 = ", Bold], f1, Style[", f2 = ", Bold],
     f2, Style[", f3 = ", Bold], f3, Style[", f4 = ", Bold], f4}],
  Row[{Style["z1 = ", Bold], FullSimplify[1 / (f1 - 1)], Style[", z2 = ", Bold],
     FullSimplify[1 / (f2 - 1)], Style[", z3 = ", Bold], FullSimplify[1 / (f3 - 1)],
     Style[", z4 = ", Bold], FullSimplify[1 / (f4 - 1)]}],
  Row[{Style["x1 = ", Bold], FullSimplify[1 / (r1 - 1)], Style[", x2 = ", Bold],
     FullSimplify[1 / (r2 - 1)], Style[", x3 = ", Bold], FullSimplify[1 / (r3 - 1)],
     Style[", x4 = ", Bold], FullSimplify[1 / (r4 - 1)]}],
  Row[{Style["y1 = ", Bold], FullSimplify[1 / (s1 - 1)], Style[", y2 = ", Bold],
     FullSimplify[1 / (s2 - 1)], Style[", y3 = ", Bold], FullSimplify[1 / (s3 - 1)],
     Style[", y4 = ", Bold], FullSimplify[1 / (s4 - 1)]}],
```

```
  Row[{Style["p1 = ", Bold], Simplify[Sqrt[r1 - 1]],
    Style[", p2 = ", Bold], Simplify[Sqrt[r2 - 1]], Style[", p3 = ", Bold],
    Simplify[Sqrt[r3 - 1]], Style[", p4 = ", Bold], Simplify[Sqrt[r4 - 1]]}],
  Row[{Style["q1 = ", Bold], Simplify[Sqrt[s1 - 1]],
    Style[", q2 = ", Bold], Simplify[Sqrt[s2 - 1]], Style[", q3 = ", Bold],
    Simplify[Sqrt[s3 - 1]], Style[", q4 = ", Bold], Simplify[Sqrt[s4 - 1]]}],
  Row[{Style["p1·q1 = ", Bold], Simplify[Sqrt[r1 - 1] Sqrt[s1 - 1]],
    Style[", p2·q2 = ", Bold], Simplify[Sqrt[r2 - 1] Sqrt[s2 - 1]],
    Style[", p3·q3 = ", Bold], Simplify[Sqrt[r3 - 1] Sqrt[s3 - 1]],
    Style[", p4·q4 = ", Bold], Simplify[Sqrt[r4 - 1] Sqrt[s4 - 1]]}],
  Row[{Style["\!\(\*OverscriptBox[\(α1\), \(_\)]\) = ", Bold],
    σ1 - anglesDeg⟦1, 1⟧, "°", Style[
     ", \!\(\*OverscriptBox[\(β1\), \(_\)]\) = ", Bold], σ1 - anglesDeg⟦1, 2⟧,
    "°", Style[", \!\(\*OverscriptBox[\(γ1\), \(_\)]\) = ", Bold],
    σ1 - anglesDeg⟦1, 3⟧, "°",
    Style[", \!\(\*OverscriptBox[\(δ1\), \(_\)]\) = ", Bold],
    σ1 - anglesDeg⟦1, 4⟧, "°"}]
 }]


(*=============================================================================
 =====*)
(*=============================
  BRICARD's EQUATIONS=============================*)
(*=============================================================================
 =====*)

(*Bricard's equation as defined in terms of angles and variables*)
BricardsEquation[{α_, β_, γ_, δ_}, σ_, x_, y_] := Module[
   {c22, c20, c02, c11, c00},
   c22 = Sin[σ - δ] Sin[σ - δ - β];
   c20 = Sin[σ - α] Sin[σ - α - β];
   c02 = Sin[σ - γ] Sin[σ - γ - β];
   c11 = -Sin[α] Sin[γ];
   c00 = Sin[σ] Sin[σ - β];
   c22 x^2 y^2 + c20 x^2 + c02 y^2 + 2 c11 x y + c00
  ];


(*Step 1: Bricard's system of equations*)
Column[{(*Header*)TextCell[
   Style["================= Bricard's System of Equations =================",
    Red, Bold, 16], "Text"], (*Explanatory note*)
  Row[
   {TextCell[Style["We introduce new notation for the cotangents of half of
        the dihedral angles. Denote w₁:= ", GrayLevel[0.3], 13],
     "Text"], TraditionalForm[cot[Subscript[θ, 1] / 2]], TextCell[
```

```
      Style[", t:= ", GrayLevel[0.3], 13], "Text"],
     TraditionalForm[cot[Subscript[θ, 2] / 2]],
     TextCell[Style[", w₂:= ", GrayLevel[0.3], 13], "Text"],
     TraditionalForm[cot[Subscript[θ, 3] / 2]],
     TextCell[Style[", and z:= ", GrayLevel[0.3], 13], "Text"],
     TraditionalForm[cot[Subscript[θ, 4] / 2]]
    }], Spacer[12],
   (*Traditional form results*)Row[{"P₁(w₁, z) = ",
     TraditionalForm[Collect[FullSimplify[BricardsEquation[anglesDeg⟦1⟧ Degree,
         sigmas⟦1⟧ Degree, w₁, z]], w₁]], " = 0"}], Spacer[6],
   Row[{"P₂(w₁, t) = ",
     TraditionalForm[Collect[FullSimplify[ BricardsEquation[anglesDeg⟦2⟧ Degree,
         sigmas⟦2⟧ Degree, w₁, t]], w₁]], " = 0"}], Spacer[6],
   Row[{"P₃(w₂, t) = ",
     TraditionalForm[Collect[FullSimplify[BricardsEquation[anglesDeg⟦3⟧ Degree,
         sigmas⟦3⟧ Degree, w₂, t]], w₂]], " = 0"}], Spacer[6],
   Row[{"P₄(w₂, z) = ", TraditionalForm[Collect[FullSimplify[BricardsEquation[
         anglesDeg⟦4⟧ Degree, sigmas⟦4⟧ Degree, w₂, z]], w₂]], " = 0"}]
  }]

(*discriminant*)
Disc[t_] := (Sin[sigma1 Degree] Sin[(sigma1 - alpha1 - beta1) Degree] +
      t^2 Sin[(sigma1 - alpha1) Degree] Sin[(sigma1 - beta1) Degree])
    (Sin[(sigma1 - gamma1) Degree] Sin[(sigma1 - delta1) Degree] + t^2 Sin[
        (sigma1 - beta1 - gamma1) Degree] Sin[(sigma1 - beta1 - delta1) Degree]);

(*==================================
 FLEXION 1==================================*)
e0 = 1;
W2[t_] := t;
Z[t_] := ( -t Sin[beta1 Degree] + e0 Sqrt[Disc[t]]) /
    (Sin[(sigma1 - delta1) Degree] Sin[(sigma1 - alpha1 - beta1) Degree] +
      t^2 Sin[(sigma1 - alpha1) Degree] Sin[(sigma1 - beta1 - delta1) Degree]);
U[t_] := ( t Sin[beta1 Degree] + e0 Sqrt[Disc[t]]) /
    (Sin[(sigma1 - delta1) Degree] Sin[(sigma1 - alpha1 - beta1) Degree] +
      t^2 Sin[(sigma1 - alpha1) Degree] Sin[(sigma1 - beta1 - delta1) Degree]);
W1[t_] := e0 Sqrt[Disc[t]] /
     (Sin[(sigma1 - gamma1) Degree] Sin[(sigma1 - delta1) Degree] +
       t^2 Sin[(sigma1 - beta1 - gamma1) Degree]
        Sin[(sigma1 - beta1 - delta1) Degree]);

(*==================================
 FLEXION 2==================================*)
e00 = -1;
W22[t_] := t;
Z2[t_] := ( -t Sin[beta1 Degree] + e00 Sqrt[Disc[t]]) /
    (Sin[(sigma1 - delta1) Degree] Sin[(sigma1 - alpha1 - beta1) Degree] +
```

```
     t^2 Sin[(sigma1 - alpha1) Degree] Sin[(sigma1 - beta1 - delta1) Degree]);
U2[t_] := ( t Sin[beta1 Degree] + e00 Sqrt[Disc[t]]) /
    (Sin[(sigma1 - delta1) Degree] Sin[(sigma1 - alpha1 - beta1) Degree] +
     t^2 Sin[(sigma1 - alpha1) Degree] Sin[(sigma1 - beta1 - delta1) Degree]);
W12[t_] := e00 Sqrt[Disc[t]] /
     (Sin[(sigma1 - gamma1) Degree] Sin[(sigma1 - delta1) Degree] +
       t^2 Sin[(sigma1 - beta1 - gamma1) Degree]
        Sin[(sigma1 - beta1 - delta1) Degree]);


(*Step 2: Checking that formula from Theorem 1 simplifies to (E.1)*)
Column[
 {(*Header*)TextCell[Style["================= FLEXIONS =================",
    Darker[Red], Bold, 16], "Text"], (*Explanatory note*)
  TextCell[Style["Solutions to Bricard's equations under a free parameter
      t ∈ ℂ using Theorem 1:", GrayLevel[0.3], 13], "Text"], Spacer[12],
  (*Heading for results*)TextCell[Style["Solution 1:", Bold, 14], "Text"],
  Spacer[6], (*Traditional form results*)
  Row[{"w₁(t) = ", TraditionalForm[FullSimplify[Z[t]]]}], Spacer[6],
  Row[{"t(t) = ", TraditionalForm[FullSimplify[W2[t]]]}], Spacer[6],
  Row[{"w₂(t) = ", TraditionalForm[FullSimplify[U[t]]]}], Spacer[6],
  Row[{"z(t) = ", TraditionalForm[FullSimplify[W1[t]]]}],  Spacer[12],
  (*Heading for results*)TextCell[Style["Solution 2:", Bold, 14], "Text"],
  Spacer[6], (*Traditional form results*)
  Row[{"w₁(t) = ", TraditionalForm[FullSimplify[Z2[t]]]}], Spacer[6],
  Row[{"t(t) = ", TraditionalForm[FullSimplify[W22[t]]]}], Spacer[6],
  Row[{"w₂(t) = ", TraditionalForm[FullSimplify[U2[t]]]}], Spacer[6],
  Row[{"z(t) = ", TraditionalForm[FullSimplify[W12[t]]]}]
 }]

(*Step 3.1: Compute and print all P_i for flexion 1*)
TextCell[
 Style["================= FLEXIBILITY (Double Checking) =================",
  Orange, Bold, 16], "Text"]
funcs = {{w₁, z}, {w₁, t}, {w₂, t}, {w₂, z}};
TextCell[Style["Solution 1:", Bold, 14], "Text"]
Do[angles = anglesDeg[[i]] Degree;
  sigma = sigmas[[i]] Degree;
  {α, β, γ, δ} = angles;
  poly = Which[i == 1, BricardsEquation[{α, β, γ, δ}, sigma, Z[t], W1[t]],
    i == 2, BricardsEquation[{α, β, γ, δ}, sigma, Z[t], W2[t]],
    i == 3, BricardsEquation[{α, β, γ, δ}, sigma, U[t], W2[t]],
    i == 4, BricardsEquation[{α, β, γ, δ}, sigma, U[t], W1[t]]];
  Print[Row[{Subscript["P", i], "(", funcs[[i, 1]],
     ", ", funcs[[i, 2]], ") = ", FullSimplify[poly]}]], {i, 1, 4}];

(*Step 3.2: Compute and print all P_i for flexion 2*)
```

```
TextCell[Style["Solution 2:", Bold, 14], "Text"]
funcs = {{w₁, z}, {w₁, t}, {w₂, t}, {w₂, z}};
Do[angles = anglesDeg⟦i⟧ Degree;
  sigma = sigmas⟦i⟧ Degree;
  {α, β, γ, δ} = angles;
  poly = Which[i == 1, BricardsEquation[{α, β, γ, δ}, sigma, Z2[t], W12[t]],
    i == 2, BricardsEquation[{α, β, γ, δ}, sigma, Z2[t], W22[t]],
    i == 3, BricardsEquation[{α, β, γ, δ}, sigma, U2[t], W22[t]],
    i == 4, BricardsEquation[{α, β, γ, δ}, sigma, U2[t], W12[t]]];
  Print[Row[{Subscript["P", i], "(", funcs⟦i, 1⟧,
     ", ", funcs⟦i, 2⟧, ") = ", FullSimplify[poly]}]], {i, 1, 4}];


(*Step 4:
 Checking that non-flexible solution solves Bricard's system of equation*)
(*=================================
 NON-FLEXIBLE SOLUTION=================================*)
W2NF = Sqrt[1 + Sqrt[3]];
ZNF = 0;
UNF = 0;
W1NF = Sqrt[3 / 2] * W2NF;

Column[{(*Header*)
  TextCell[Style["================= NON-FLEXIBLE SOLUTION =================",
    Darker[Orange], Bold, 16], "Text"], (*Explanatory note*)TextCell[
   Style["Another solutions to Bricard's equations (non-flexible solution):",
    GrayLevel[0.3], 13], "Text"], Spacer[12],
  Row[{"w₁ = ", TraditionalForm[FullSimplify[ZNF]]}], Spacer[6],
  Row[{"t = ", TraditionalForm[FullSimplify[W2NF]]}], Spacer[6],
  Row[{"w₂ = ", TraditionalForm[FullSimplify[UNF]]}], Spacer[6],
  Row[{"z = ", TraditionalForm[FullSimplify[W1NF]]}]
 }]

(*Compute and print all P_i for non-flexible solution*)
TextCell[
 Style["================= Bricard's System of Equation with NON-Flexible
    Solution =================", Darker[Cyan], Bold, 16], "Text"]
funcs = {{w₁, z}, {w₁, t}, {w₂, t}, {w₂, z}};
TextCell[Style["Non-Flexible Solution:", Bold, 14], "Text"]
Do[angles = anglesDeg⟦i⟧ Degree;
  sigma = sigmas⟦i⟧ Degree;
  {α, β, γ, δ} = angles;
  poly = Which[i == 1, BricardsEquation[{α, β, γ, δ}, sigma, ZNF, W1NF],
    i == 2, BricardsEquation[{α, β, γ, δ}, sigma, ZNF, W2NF],
    i == 3, BricardsEquation[{α, β, γ, δ}, sigma, UNF, W2NF],
    i == 4, BricardsEquation[{α, β, γ, δ}, sigma, UNF, W1NF]];
  Print[Row[{Subscript["P", i], "(", funcs⟦i, 1⟧,
     ", ", funcs⟦i, 2⟧, ") = ", FullSimplify[poly]}]], {i, 1, 4}];
```

Out[793]=

================= CONDITION (N.0) =================

✔ All vertices satisfy (N.0).

Out[796]=

================= CONDITION (N.3) =================

✔ M1 = M2 = M3 = M4 = $-1+\sqrt{3}$

Out[802]=

================= CONDITION (N.4) =================

✔ r1 = r2 = $4-2\sqrt{3}$; ✔ r3 = r4 = $4-2\sqrt{3}$

✔ s1 = s4 = $1+\sqrt{3}$; ✔ s2 = s3 = $2-\sqrt{3}$

Out[812]=

================= CONDITION (N.5) =================

⚠ *Approximate validation using*
ε-tolerance. For rigorous proof, see the referenced paper.

✅ Valid Combination Found (M < 1):

e1 = −1, e2 = −1, e3 = 1

t1 = 1.K + 0.600443iK'

t2 = 2.K + 0.600443iK'

t3 = 0.K + 0.600443iK'

t4 = 1.K + 0.600443iK'

t1 + e1*t2 + e2*t3 + e3*t4 = 0.K + -8.88178×10^{-16}iK'

Out[814]=

================= OTHER PARAMETERS =================

u = $2-\sqrt{3}$

σ1 = 120 °, σ2 = 120 °, σ3 = 240 °, σ4 = 120 °

f1 = $\frac{1}{2}\left(-1+\sqrt{3}\right)$, f2 = 2, f3 = 2, f4 = $\frac{1}{2}\left(-1+\sqrt{3}\right)$

z1 = $-1-\frac{1}{\sqrt{3}}$, z2 = 1, z3 = 1, z4 = $-1-\frac{1}{\sqrt{3}}$

x1 = $-1-\frac{2}{\sqrt{3}}$, x2 = $-1-\frac{2}{\sqrt{3}}$, x3 = $-1-\frac{2}{\sqrt{3}}$, x4 = $-1-\frac{2}{\sqrt{3}}$

y1 = $\frac{1}{\sqrt{3}}$, y2 = $\frac{1}{1-\sqrt{3}}$, y3 = $\frac{1}{1-\sqrt{3}}$, y4 = $\frac{1}{\sqrt{3}}$

p1 = $i\sqrt{-3+2\sqrt{3}}$, p2 = $i\sqrt{-3+2\sqrt{3}}$, p3 = $i\sqrt{-3+2\sqrt{3}}$, p4 = $i\sqrt{-3+2\sqrt{3}}$

q1 = $3^{1/4}$, q2 = $i\sqrt{-1+\sqrt{3}}$, q3 = $i\sqrt{-1+\sqrt{3}}$, q4 = $3^{1/4}$

p1·q1 = $i\,3^{1/4}\sqrt{-3+2\sqrt{3}}$, p2·q2 = $-\sqrt{9-5\sqrt{3}}$

, p3·q3 = $-\sqrt{9-5\sqrt{3}}$, p4·q4 = $i\,3^{1/4}\sqrt{-3+2\sqrt{3}}$

$\overline{\alpha 1}$ = 105°, $\overline{\beta 1}$ = 60°, $\overline{\gamma 1}$ = 45°, $\overline{\delta 1}$ = 30°

Out[816]=

================= Bricard's System of Equations =================

We introduce new notation for the cotangents of half of the dihedral angles. Denote $w_1 := \cot\left(\frac{\theta_1}{2}\right)$, $t := \cot\left(\frac{\theta_2}{2}\right)$, $w_2 := \cot\left(\frac{\theta_3}{2}\right)$, and $z := \cot\left(\frac{\theta_4}{2}\right)$

$$P_1(w_1, z) = \frac{1}{4} w_1^2 \left(-z^2 + \sqrt{3} + 1\right) - \frac{w_1 z}{2} + \frac{1}{4}\left(-\sqrt{3}\, z^2 + z^2 + 3\right) = 0$$

$$P_2(w_1, t) = \frac{1}{8}\left(\sqrt{2}\left(1 + \sqrt{3}\right) t^2 - 2\sqrt{2}\right) w_1^2 + \frac{1}{8}\left(\sqrt{6}\, t^2 - 3\sqrt{2}\, t^2 + 2\sqrt{6}\right) - \sqrt{3}\, t\, w_1 = 0$$

$$P_3(w_2, t) = \frac{1}{8}\left(2\sqrt{2} - \sqrt{2}\left(1 + \sqrt{3}\right) t^2\right) w_2^2 + \frac{1}{8}\left(-\sqrt{6}\, t^2 + 3\sqrt{2}\, t^2 - 2\sqrt{6}\right) - \sqrt{3}\, t\, w_2 = 0$$

$$P_4(w_2, z) = \frac{1}{4} w_2^2 \left(-z^2 + \sqrt{3} + 1\right) - \frac{w_2 z}{2} + \frac{1}{4}\left(-\sqrt{3}\, z^2 + z^2 + 3\right) = 0$$

Out[828]=

================= FLEXIONS =================

Solutions to Bricard's equations under a free parameter $t \in \mathbb{C}$ using Theorem 1:

Solution 1:

$$w_1(t) = \frac{2\sqrt{6}\, t - \sqrt{2}\sqrt{\sqrt{3}\, t^4 + 6 t^2 + 2\sqrt{3}}}{\left(1 + \sqrt{3}\right) t^2 - 2}$$

$$t(t) = t$$

$$w_2(t) = -\frac{\sqrt{2}\left(\sqrt{\sqrt{3}\, t^4 + 6 t^2 + 2\sqrt{3}} + 2\sqrt{3}\, t\right)}{\left(1 + \sqrt{3}\right) t^2 - 2}$$

$$z(t) = \frac{\sqrt{2}\sqrt{\sqrt{3}\, t^4 + 6 t^2 + 2\sqrt{3}}}{\left(\sqrt{3} - 1\right) t^2 + 2}$$

Solution 2:

$$w_1(t) = \frac{\sqrt{2}\left(\sqrt{\sqrt{3}\, t^4 + 6 t^2 + 2\sqrt{3}} + 2\sqrt{3}\, t\right)}{\left(1 + \sqrt{3}\right) t^2 - 2}$$

$$t(t) = t$$

$$w_2(t) = \frac{\sqrt{2}\left(\sqrt{\sqrt{3}\, t^4 + 6 t^2 + 2\sqrt{3}} - 2\sqrt{3}\, t\right)}{\left(1 + \sqrt{3}\right) t^2 - 2}$$

$$z(t) = -\frac{\sqrt{2}\sqrt{\sqrt{3}\, t^4 + 6 t^2 + 2\sqrt{3}}}{\left(\sqrt{3} - 1\right) t^2 + 2}$$

Out[829]=

================= FLEXIBILITY (Double Checking) =================

Out[831]=

Solution 1:

$P_1(w_1, z) = 0$

$P_2(w_1, t) = 0$

$P_3(w_2, t) = 0$

$P_4(w_2, z) = 0$

Out[833]=

Solution 2:

$P_1(w_1, z) = 0$

$P_2(w_1, t) = 0$

$P_3(w_2, t) = 0$

$P_4(w_2, z) = 0$

Out[840]=

================= NON-FLEXIBLE SOLUTION =================

Another solutions to Bricard's equations (non-flexible solution):

$w_1 = 0$

$t = \sqrt{1 + \sqrt{3}}$

$w_2 = 0$

$z = \sqrt{\frac{3}{2}\left(1 + \sqrt{3}\right)}$

Out[841]=

================= Bricard's System of Equation with NON-Flexible Solution =================

Out[843]=

Non-Flexible Solution:

$P_1(w_1, z) = 0$

$P_2(w_1, t) = 0$

$P_3(w_2, t) = 0$

$P_4(w_2, z) = 0$