

AE_a2x3

Eine exakte Autoevolute mit 6 Tetraedern, also 12 Punkten.

(Es wird genau ein Exemplar eines Tetraeders sechs mal geklont. Der Diederwinkel des Tetraeders liegt bei $\cos(\Phi)=1/3$, das ist die Einparameter-Familie in der auch das Platonische Vierflach liegt.)

Die äußeren Symmetrien der AE entsprechen $D_3 = Z_2 \times Z_3$.

Das sind ..

1. die Spiegelung an der x-Achse
2. die Drehung um $2\pi/3$ um die z-Achse

In[1]:=

```
(* Die Drehung um 2 Pi/3 um die z-Achse, Multiplikation von rechts. *)
drehz120 = {{-1/2, Sqrt[3]/2, 0}, {-Sqrt[3]/2, -1/2, 0}, {0, 0, 1}};
```

Der hier gesetzte Parameter t ist im Grunde der Tangens eines halben Winkels, also $t = \tan(\phi/2)$.

Daraus machen wir $\cos(\phi)$, $\sin(\phi)$ und die weiteren Parameter und Koordinaten.

Alle diese Werte liegen dann in einer algebraischen Erweiterung der rationalen Zahlen exakt vor.

In[2]:=

```
t = Sqrt[6 - 4 Sqrt[3] + Sqrt[13] (2 - Sqrt[3])];
cosphi = Simplify[(1 - t^2)/(1 + t^2)];
sinphi = Simplify[2 t/(1 + t^2)];
Print[" t = ", t]
Print[" cosphi = ", cosphi]
Print[" sinphi = ", sinphi]
```

$$t = \sqrt{6 - 4\sqrt{3} + \sqrt{13}(2 - \sqrt{3})}$$

$$\cos\phi = \frac{5 - 4\sqrt{3} + 2\sqrt{13} - \sqrt{39}}{-7 + 4\sqrt{3} - 2\sqrt{13} + \sqrt{39}}$$

$$\sin\phi = \frac{2\sqrt{6 - 4\sqrt{3} + 2\sqrt{13} - \sqrt{39}}}{-7 + 4\sqrt{3} - 2\sqrt{13} + \sqrt{39}}$$

In[8]:=

```

(* Zwei weitere abgeleitete Größen a und b *)
tmpnom = Simplify[2 + Sqrt[3] + 4 t^2 - (-2 + Sqrt[3]) t^4];
tmpden = Simplify[12 (2 - Sqrt[3] + (2 + Sqrt[3]) t^4)];
a = Simplify[Sqrt[tmpnom / tmpden]];
b = Simplify[(2 - Sqrt[3] + (2 + Sqrt[3]) t^2) / (1 + t^2)];
Print[" a = ", a]
Print[" b = ", b]

```

$$a = \frac{1}{2} \sqrt{\frac{338 - 195 \sqrt{3} + 94 \sqrt{13} - 54 \sqrt{39}}{78 - 39 \sqrt{3} + 18 \sqrt{13} - 12 \sqrt{39}}}$$

$$b = \frac{2 - 3 \sqrt{3} + \sqrt{13}}{7 - 4 \sqrt{3} + 2 \sqrt{13} - \sqrt{39}}$$

Die Koordinaten der 12 Punkte der Kurve.

Die Indizes sind modulo 12 zu lesen.

Die "innere" Symmetrie (der Indizes) ist..

1. die Spiegelung an der x-Achse: $j \rightarrow -j$
2. die Drehung um $2\pi/3$ um die z-Achse: $j \rightarrow j+8$

In[14]:=

```

ClearAll[p]
lösche alle

(* Die 4 Basispunkte, in etwa längs der x-Achse *)
p[0] = {a + 1/2, 0, 0};
p[6] = {a - 1/2, 0, 0};
p[3] = {-ab, cosphi/2, -sinphi/2};
p[9] = {-ab, -cosphi/2, sinphi/2};
(* Die erste Drehung *)
p[8] = p[0].drehz120;
p[2] = p[6].drehz120;
p[11] = p[3].drehz120;
p[5] = p[9].drehz120;
(* Die zweite Drehung *)
p[4] = p[8].drehz120;
p[10] = p[2].drehz120;
p[7] = p[11].drehz120;
p[1] = p[5].drehz120;
(* Ansonsten gilt.. (mod 12) *)
p[n_] := p[n] = p[Mod[n, 12]]
Modulo

```

Erster Test, ob die Abstände stimmen.

In[28]:=

```

(* Der Abstand (zum Quadrat) *)
distsq[x_, y_] := Simplify[(x - y).(x - y)]
vereinfache

```

In[29]:=

```

(* Die Normalen (p[j], p[j+6]) sind qua Konstruktion in Ordnung, dennoch.. *)
Print[Union[Table[1 == distsq[p[j], p[j + 6]], {j, 0, 5}]]]
gib aus Vereini... Tabelle

```

```
{True}
```

In[30]:=

```

(* Die anderen 12 "Diagonalen" gehen in einem
Rutsch. Auch da wären nur die ersten 4 zu prüfen, egal.. *)
Print[Union[Table[1 == distsq[p[j], p[j + 7]], {j, 0, 11}]]]
gib aus Vereini... Tabelle

```

```
{True}
```

Zweiter Test, ob der Evolutenpunkt stimmt, also die Kolinearität von 3 Punkten mit dem Evolutenpunkt.

In[31]:=

```
( * Der Spat: Überall ne 1 ranhängen und dann die Determinate nehmen. * )
spat[v1_, v2_, v3_, v4_] := Block[
  |evaluiere mit lokalen Variablen
  {m = {Append[v1, 1], Append[v2, 1], Append[v3, 1], Append[v4, 1]}}, Det[m]]
  |hänge an |hänge an |hänge an |hänge an |Determinante
```

In[32]:=

```
( * Hat der Spat den Wert 0? Alle 12
  in einem Lauf. Auch da wären 4 genug, egal.. * )
Print[Union[
  |gib aus |Vereinigung
  Table[Simplify[0 == spat[p[j - 1], p[j], p[j + 1], p[j + 6]]], {j, 0, 11}]]]
  |Tabelle |vereinfache
{True}
```

Numerische Werte für die Koordinaten und Bilder.

In[33]:=

```
SetOptions[Graphics3D, Axes → True,
  |setze Optionen |3D-Graphik |Achsen |wahr
  AxesLabel → {x, y, z}, AxesOrigin → {0, 0, 0}, Ticks → None];
  |Achsenbeschriftungen |Achsenursprung |Teilstriche |keine
```

In[34]:=

```
( * Obacht, coo=Table[] ist in Mathematica "klassisch" 1-basiert! * )
  |Tabelle
coo = N[Table[p[j], {j, 0, 11}]];
  |Tabelle
( * Einer mehr schadet nicht * )
AppendTo[coo, coo[[1]]];
  |hänge an bei
( * Rote Normalen * )
normals = {Thick, Red, Line[Table[{coo[[j]], coo[[j + 6]]}, {j, 6}]]];
  |dick |rot |Linie |Tabelle
```

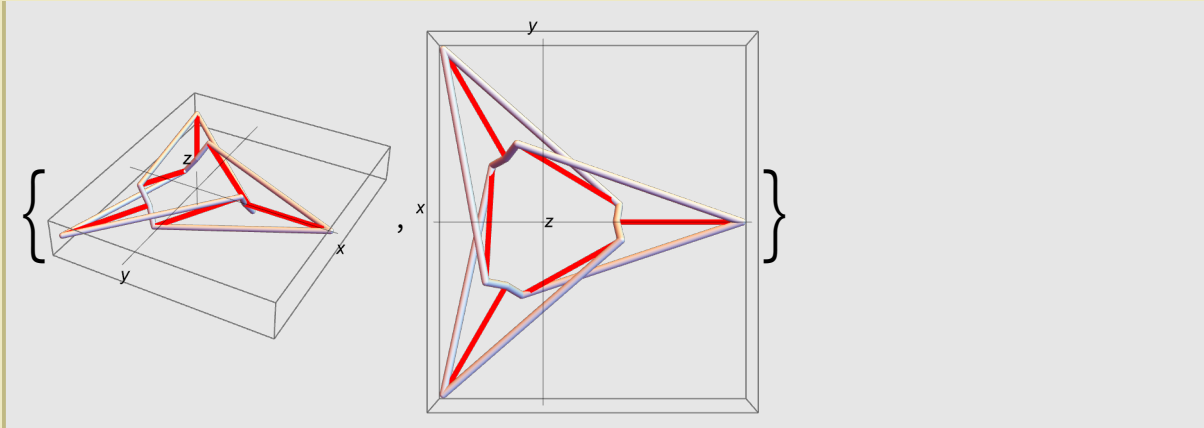
In[37]:=

```

(* Als Tube *)
  Röhre
Table[Graphics3D[{Tube[coo, .03], normals}], ViewPoint -> v],
  Tabelle 3D-Graphik Röhre Ansichtspunkt
  {v, {{1.3, -2.4, 2.}, Below}}]
  unterhalb

```

Out[37]=



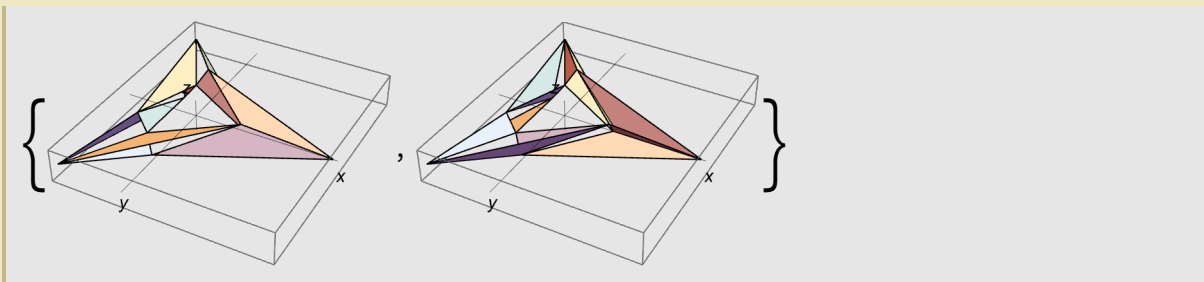
In[38]:=

```

(* Das ist nur je die eine Hälfte der Dreiecke,
jedes Tetraeder fehlen noch 2 Dreiecke *)
poly1 = Table[{Mod[j, 12, 1], Mod[j + 6, 12, 1], Mod[j + 1, 12, 1]}, {j, 1, 12}];
  Tabelle Modulo Modulo Modulo
poly2 = Table[{Mod[j, 12, 1], Mod[j + 7, 12, 1], Mod[j + 1, 12, 1]}, {j, 1, 12}];
  Tabelle Modulo Modulo Modulo
{Graphics3D[Polygon[coo, poly1]], Graphics3D[Polygon[coo, poly2]]}
  3D-Graphik Vieleck 3D-Graphik Vieleck

```

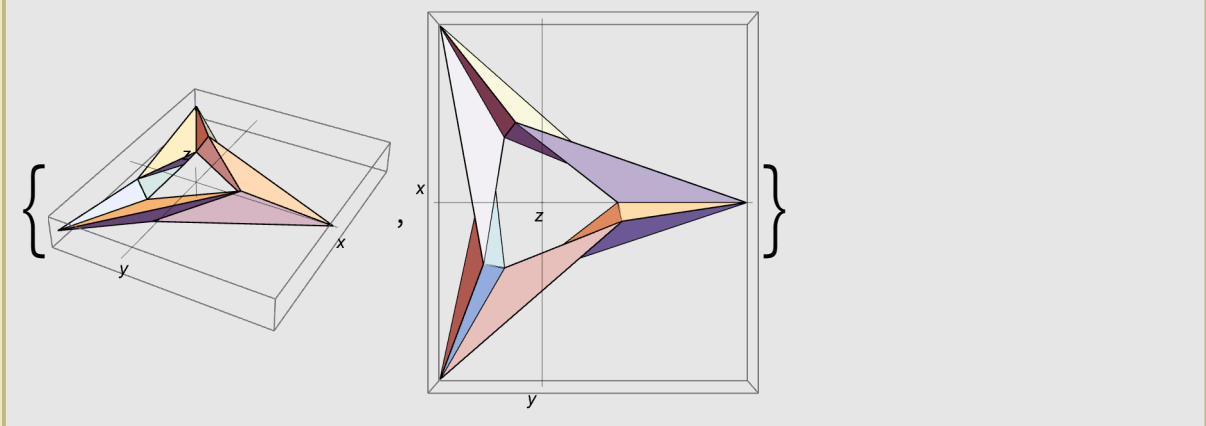
Out[40]=



In[41]:=

```
(* Als Tetraederring *)
Table[Graphics3D[Polygon[coo, Union[poly1, poly2]], ViewPoint -> v],
  {v, {{1.3, -2.4, 2.}, Below}}]
```

Out[41]=



In[42]:=

```
(* Die numerischen Koordinaten *)
Print[MatrixForm[coo]]
```

```

(
  1.58824      0.      0.
 -0.186761  0.603486  0.187573
 -0.294118  0.509428  0.
 -0.429253  0.463483 -0.187573
 -0.794118 -1.37545  0.
  0.616015 -0.140003  0.187573
  0.588236  0.      0.
  0.616015  0.140003 -0.187573
 -0.794118  1.37545  0.
 -0.429253 -0.463483  0.187573
 -0.294118 -0.509428  0.
 -0.186761 -0.603486 -0.187573
  1.58824      0.      0.
)
```