# Polyeder-Werkstatt mit *Mathematica*

*Eine Illustration der Möglichkeiten auf der Basis der Theorie im Script über Architekturmaterial, abrufbar über die Home-Page des Autors*

Script vgl. unter ===> http://www.hta-bi-bfh.ch/~wir/Scripts/Scripts.html

*von Rolf Wirz*

## A. Generieren von Polytopen mit dem *Mathematica*-Package

In[1]:=
```
Off[General::spell]
```

---

### Package laden zu Polyedern und Sternformen und anwenden

In[2]:=
```
<<Graphics`Polyhedra`
```

In[3]:=
```
?*hedoron
```

```
Information::nomatch : No symbol matching *hedoron found. Mehr…
```

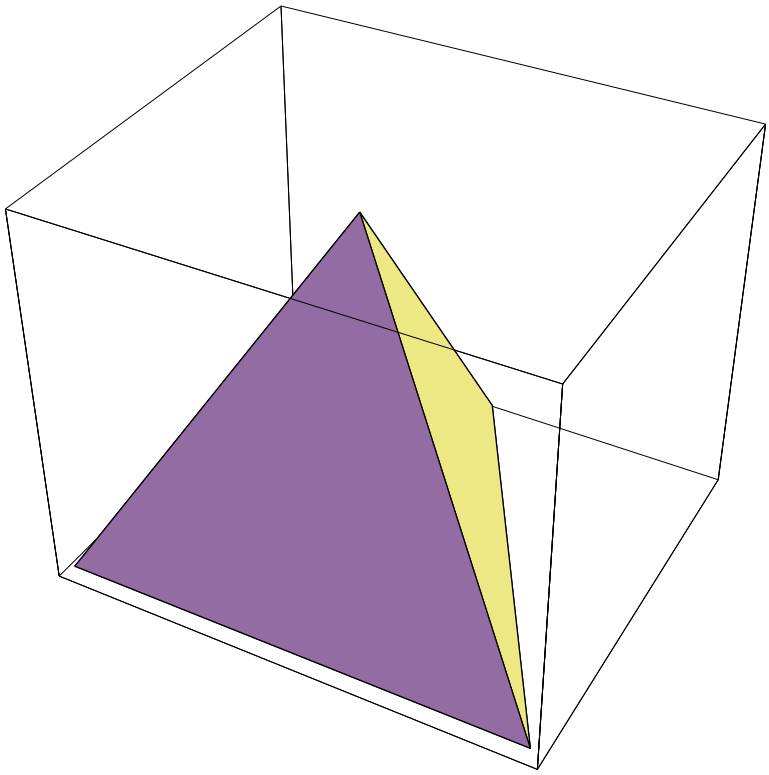
In[4]:=
```
?Graphics`Polyhedra`*
```

**Graphics`Polyhedra`**

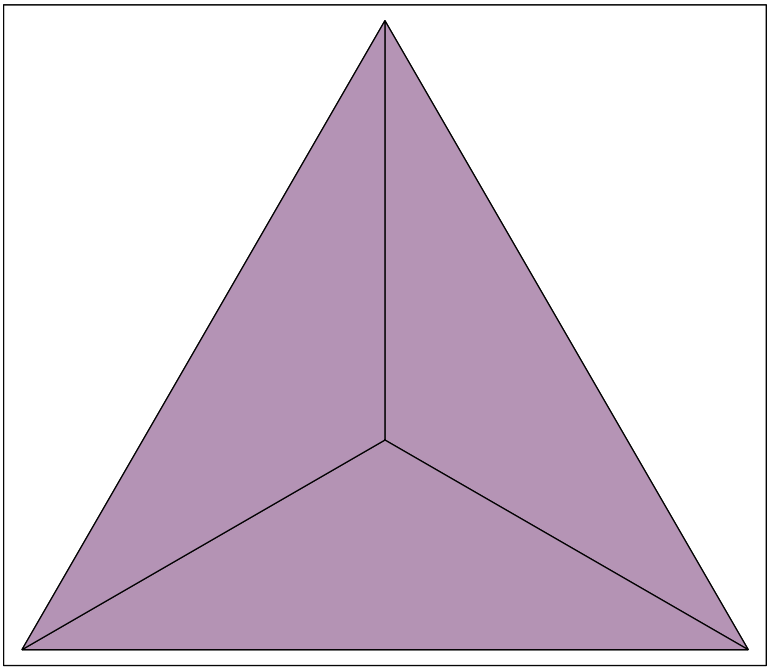
| | | |
|---|---|---|
| Geodesate | OpenTruncate | Stellate |
| GreatDodecahedron | Polyhedra | Truncate |
| GreatIcosahedron | Polyhedron | |
| GreatStellatedDodecahedron | SmallStellatedDodecahedron | |

## Tetraeder

In[5]:=
```
Show[Polyhedron[Tetrahedron]];
```

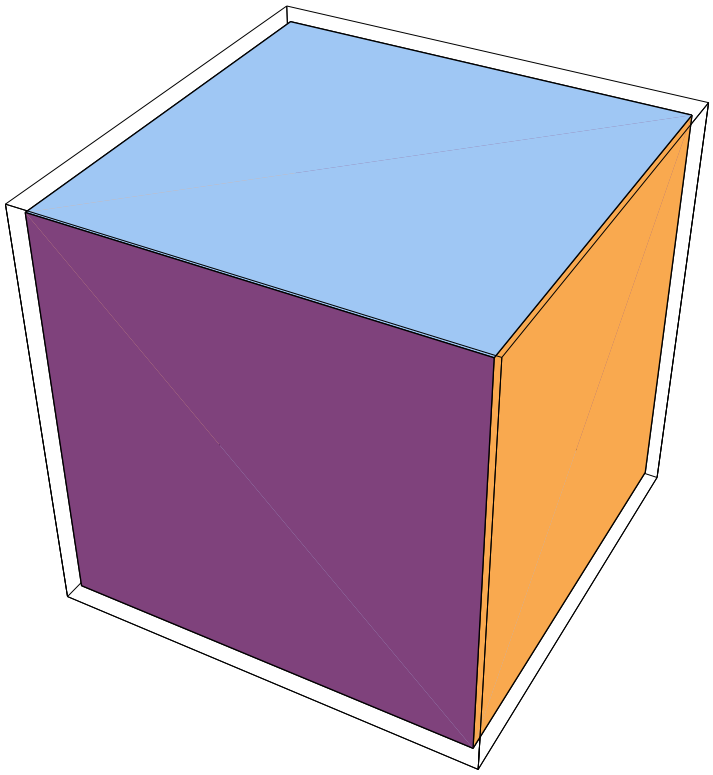


In[6]:=
```
Show[Polyhedron[Tetrahedron], ViewPoint -> {-0.0, -0.0, 1000}];
```

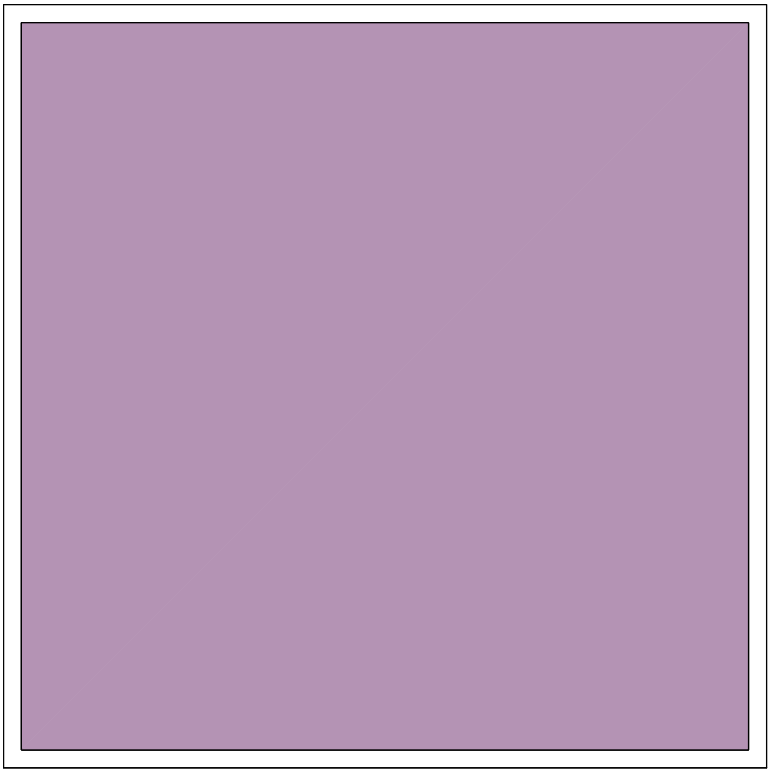


## Hexaeder

In[7]:=
```
Show[Polyhedron[Hexahedron]];
```

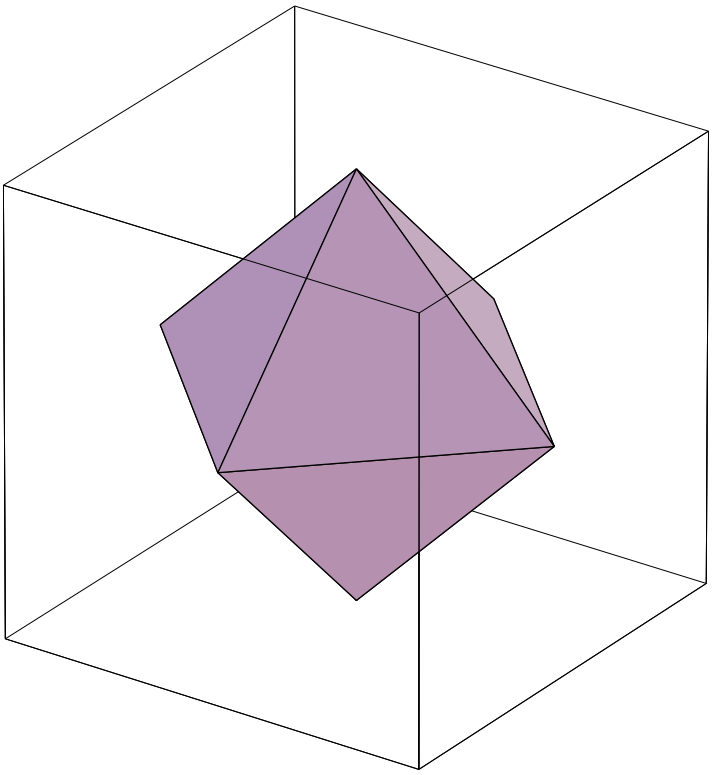


In[8]:=
```
Show[Polyhedron[Hexahedron], ViewPoint -> {-0.0, -0.0, 1000}];
```

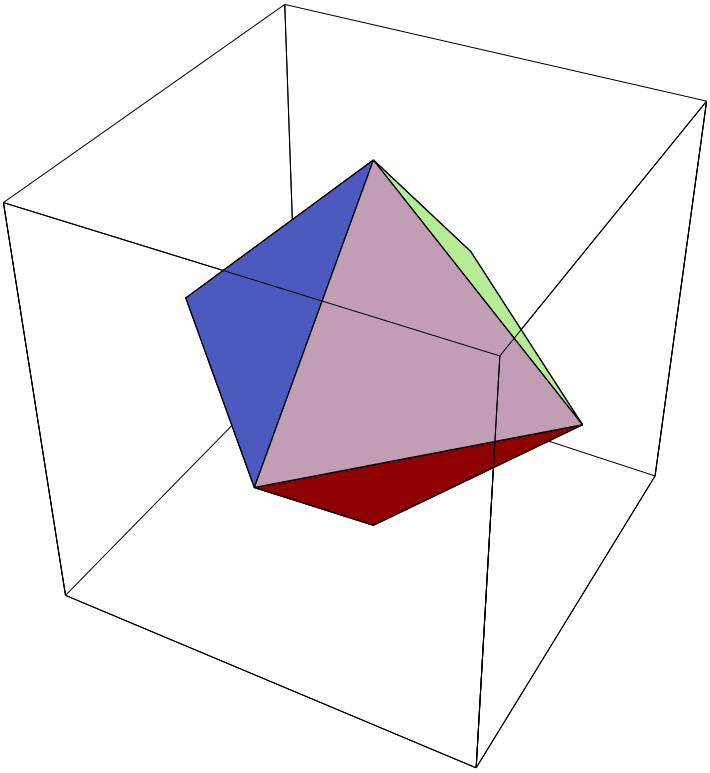


## Oktaeder

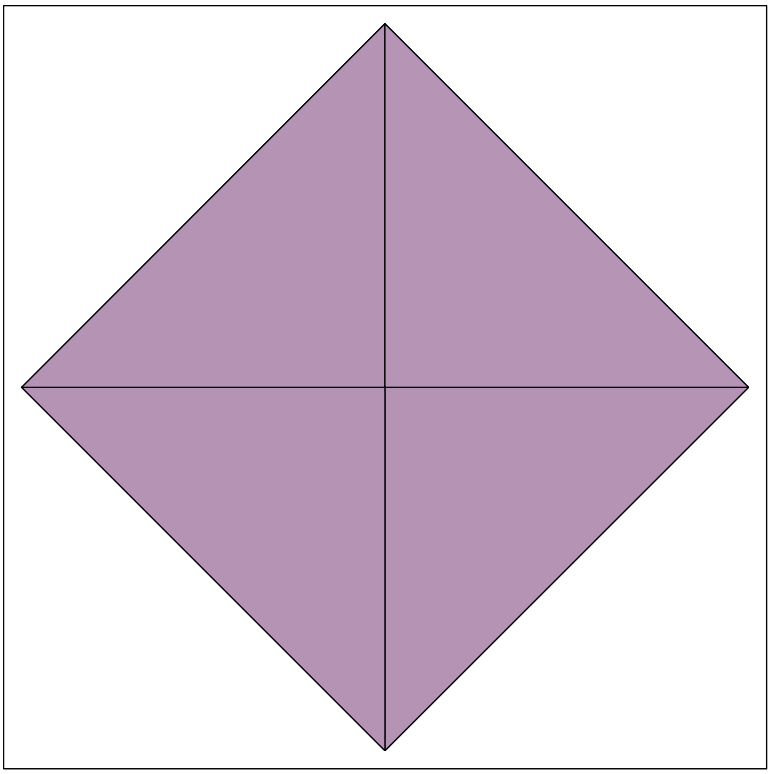
In[9]:=
```
Show[Polyhedron[Octahedron], ViewPoint -> {50, 35, 30}];
```



In[10]:=
```
Show[Polyhedron[Octahedron]];
```

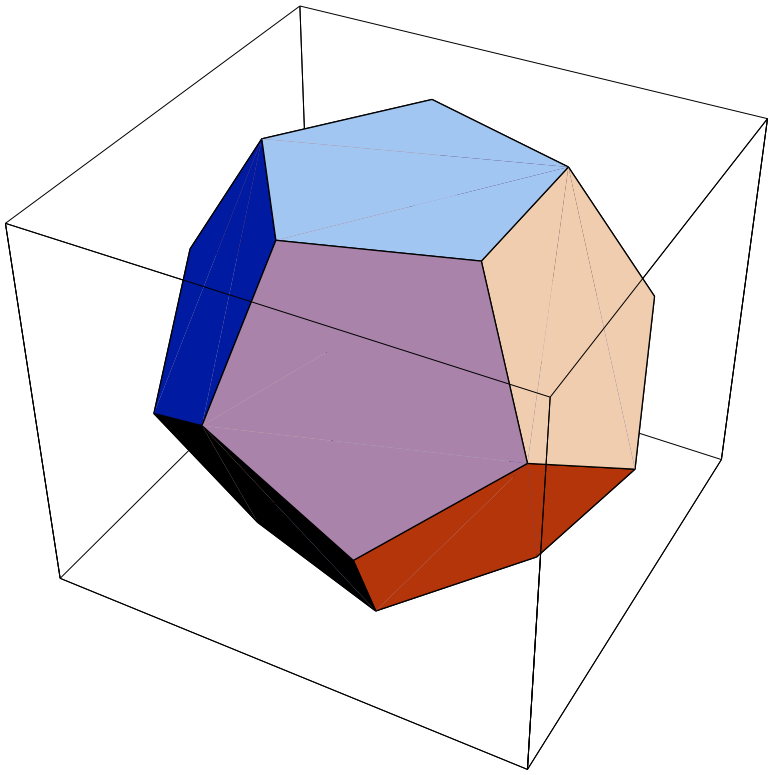


In[11]:=
```
Show[Polyhedron[Octahedron], ViewPoint -> {-0.0, -0.0, 1000}];
```

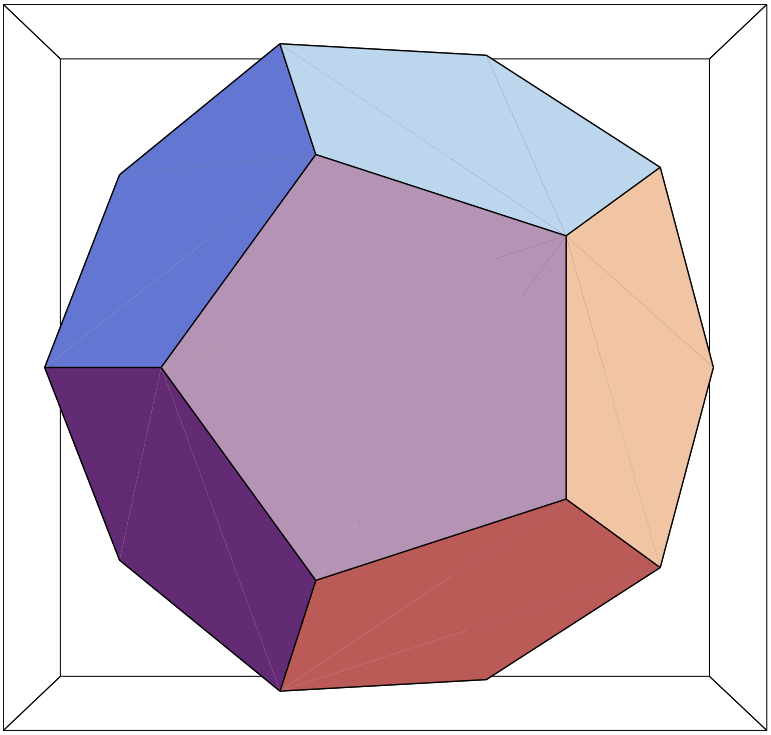


## Dodekaeder

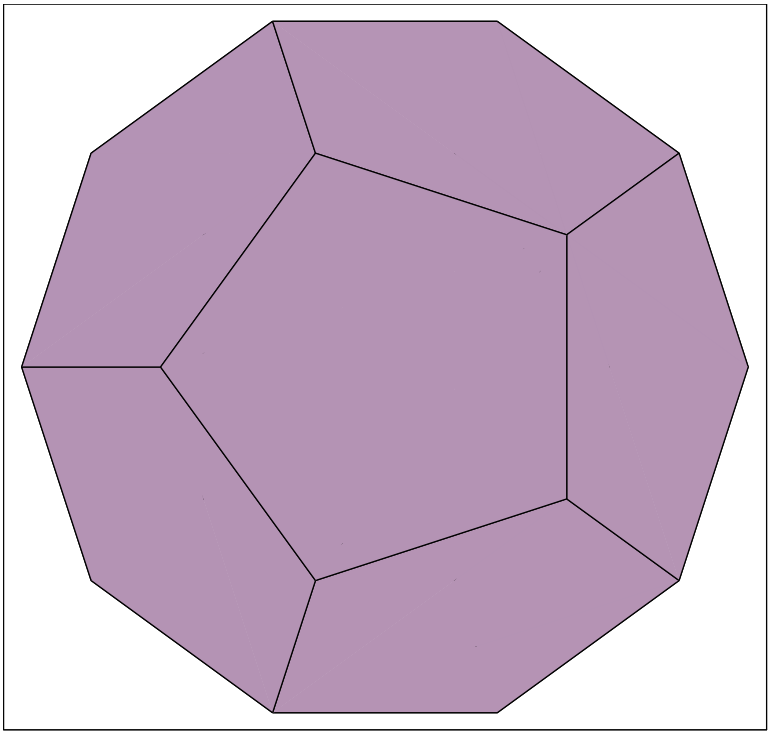
In[12]:=
```
Show[Polyhedron[Dodecahedron]];
```



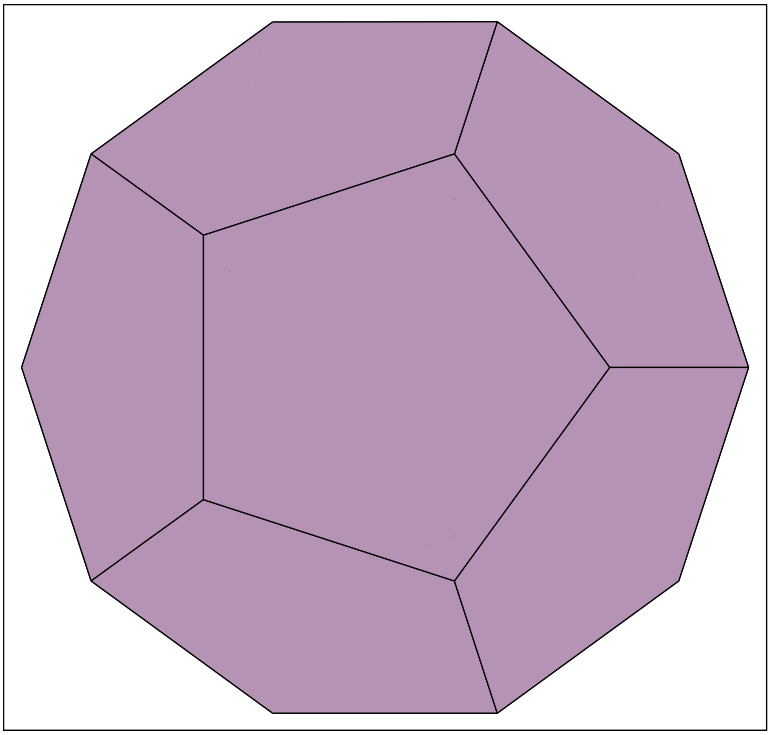
In[13]:=
```
Show[Polyhedron[Dodecahedron], ViewPoint -> {-0.0, -0.0, 5}];
```



In[14]:=
```
Show[Polyhedron[Dodecahedron], ViewPoint -> {0.0, 0.0, 1000}];
```

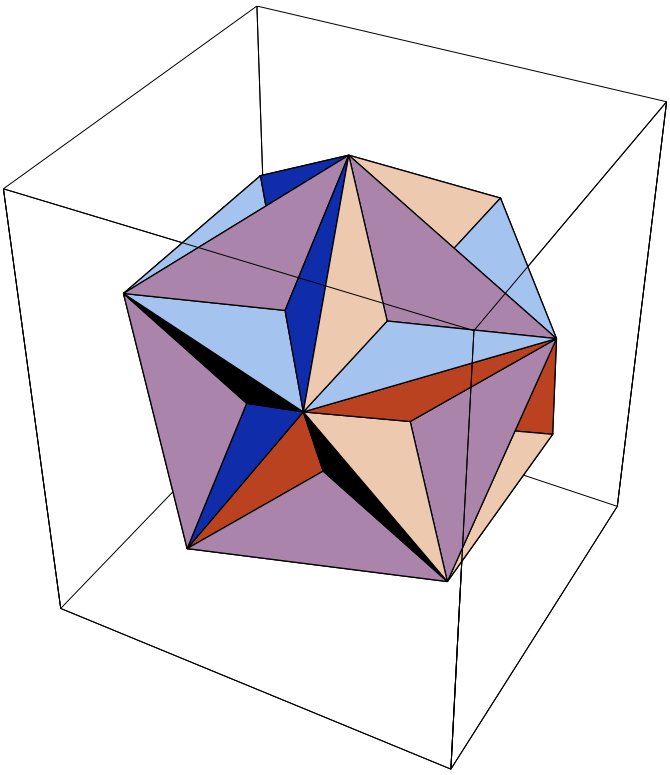


In[15]:=
```
Show[Polyhedron[Dodecahedron], ViewPoint -> {0.0, 0.0, -1000}];
```



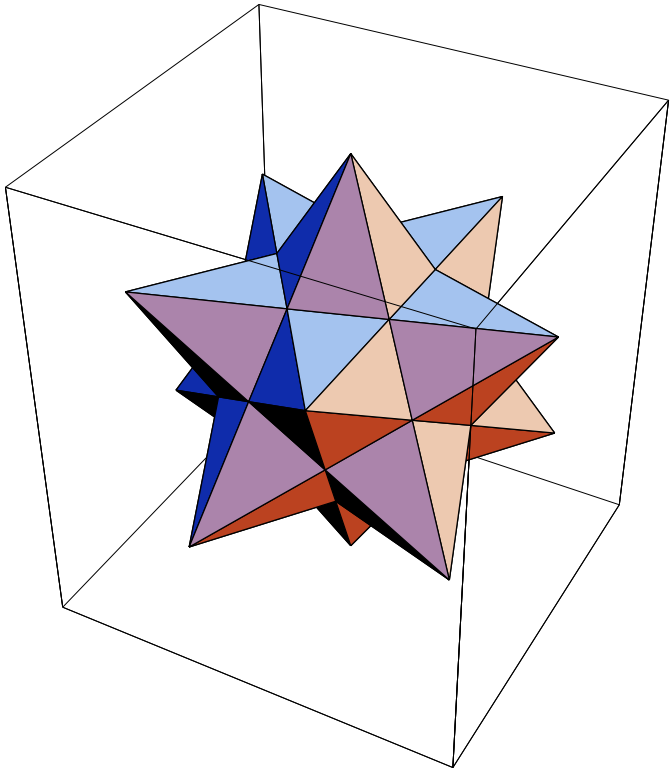
## ■ Umhülltes Ikosaeder (Dodekaederstern ???)

In[16]:=
```
Show[Polyhedron[GreatDodecahedron]];
```



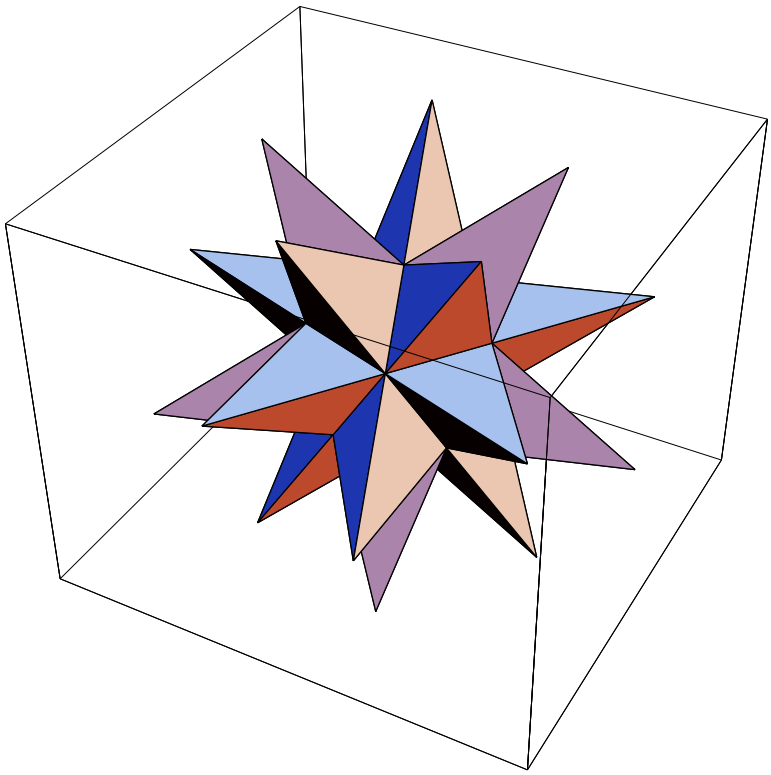
## ■ Keplerstern

In[17]:=
```
Show[Polyhedron[SmallStellatedDodecahedron]];
```



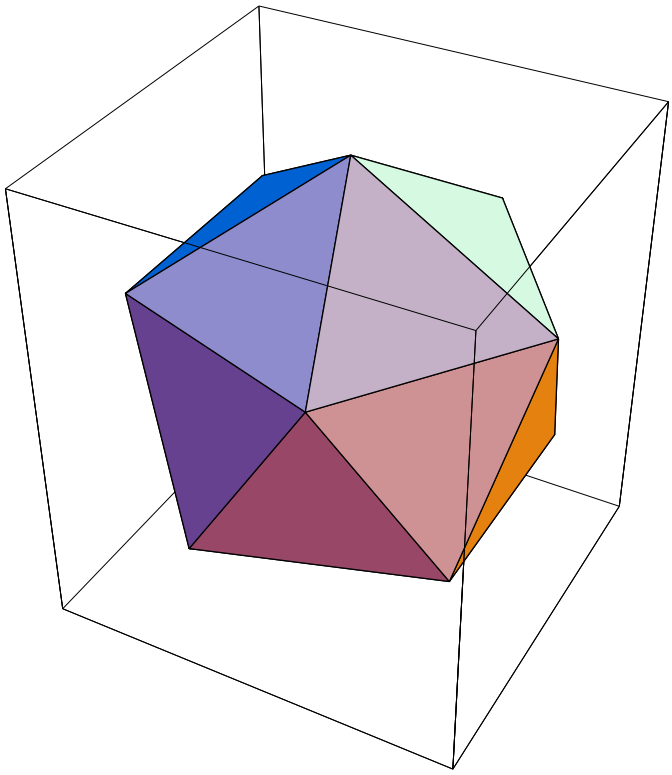
## ■ Ikosaederstern ???

In[18]:=
```
Show[Polyhedron[GreatStellatedDodecahedron]];
```

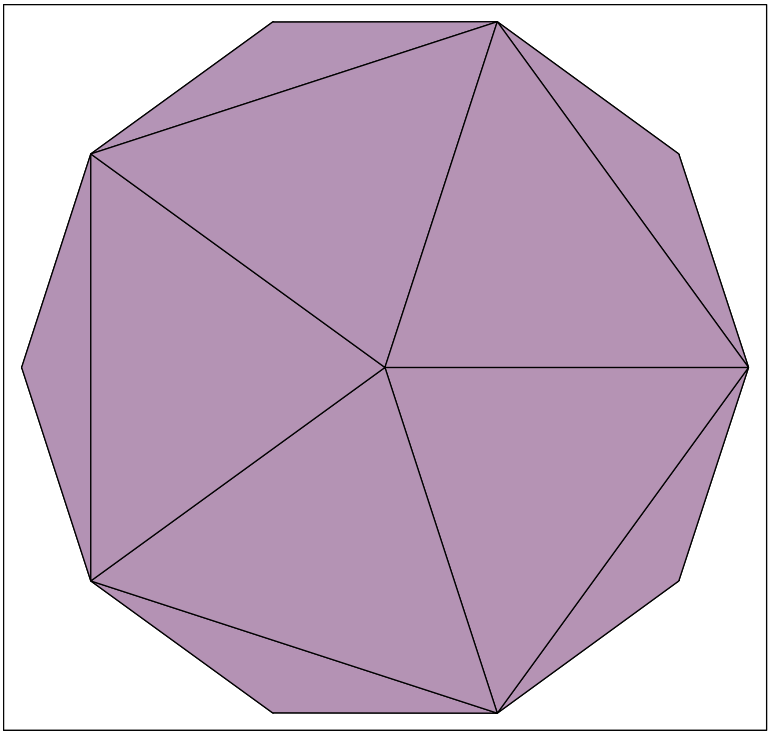


## Ikosaeder

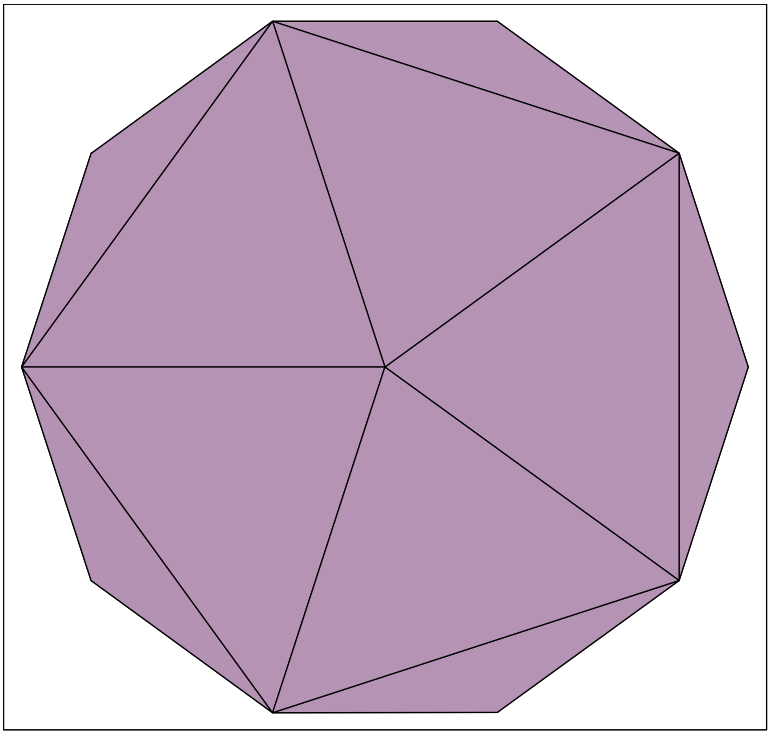
In[19]:=
```
Show[Polyhedron[Icosahedron]];
```



In[20]:=
```
Show[Polyhedron[Icosahedron], ViewPoint -> {0.0, 0.0, 1000}];
```

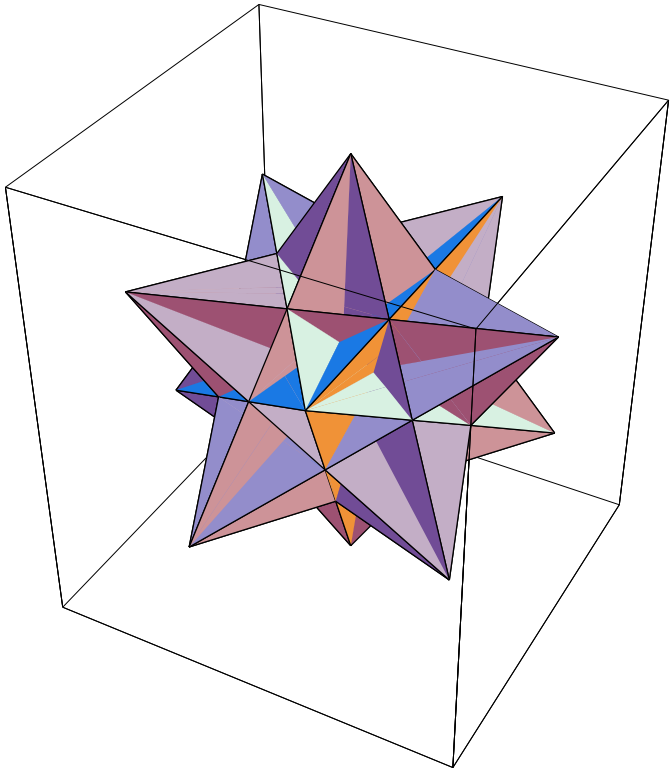


In[21]:=
```
Show[Polyhedron[Icosahedron], ViewPoint -> {0.0, 0.0, -1000}];
```



## ■ Poinsots Vielflächenstern

In[22]:=
```
Show[Polyhedron[GreatIcosahedron]];
```



■ **? Bindelstern ???**

■ **? Paravallestern ???**

■ **? Tetraederfümfling ???**

■ **? Weitere Fümflinge ???**

■ **? Hüllsterne ???**

## Spezialitäten und Animationen

In[23]:=
```
? Geodesate
```

```
Geodesate[expr, n] replaces each polygon in expr
   by the projection onto the circumscribed sphere of the
   order n regular tessellation of that polygon.  Geodesate[
   expr, n, {x, y, z}, radius] does the projection onto
   the sphere of size radius centered at {x, y, z}. Mehr…
```

In[24]:=
```
? Stellate
```

```
Stellate[expr, (ratio:2)] replaces each polygon
   in expr by a pyramid with the polygon as its base.
   Stellation ratios less than 1 give concave figures. Mehr…
```

In[25]:=
```
? Truncate
```

```
Truncate[expr] truncates each edge of
   each polygon in expr. Truncate[expr, ratio] truncates
   to the specified ratio of the edge length. Mehr…
```

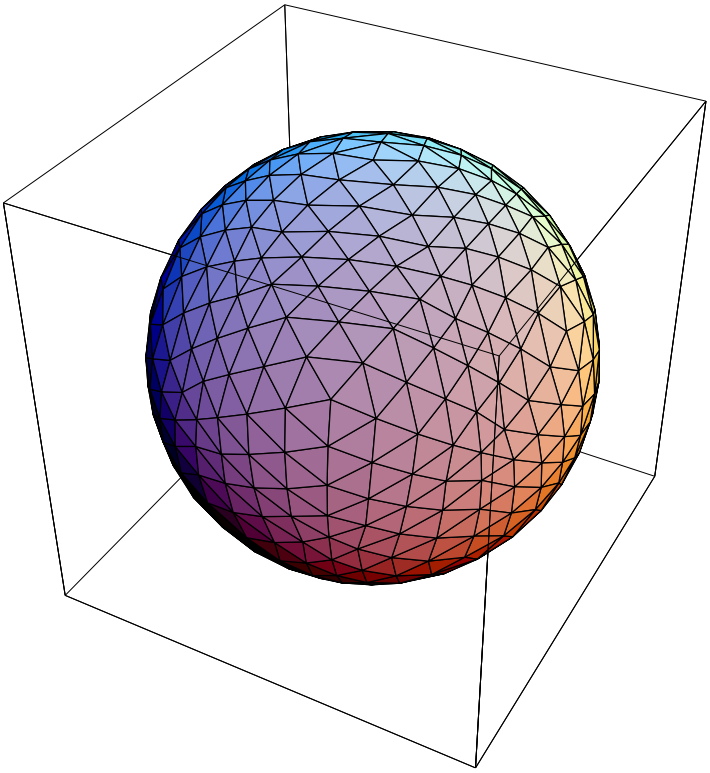
In[26]:=
```
? OpenTruncate
```

```
OpenTruncate[expr] truncates each edge of each polygon in expr
   without filling in with a polygon. OpenTruncate[expr, ratio]
   truncates to the specified ratio of the edge length. Mehr…
```

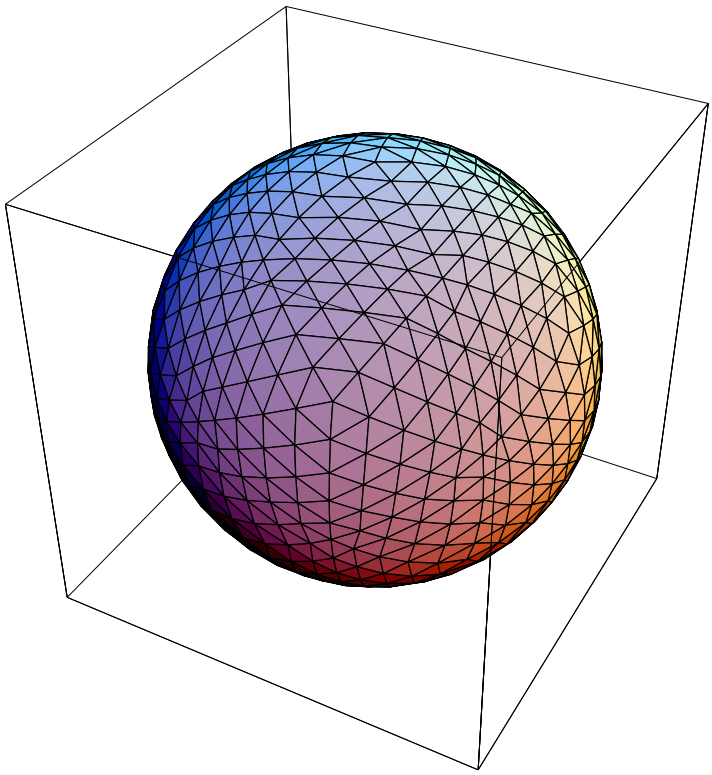
In[27]:=
```
?*hedoron
```

```
Information::nomatch : No symbol matching *hedoron found. Mehr…
```

In[28]:=
```
Show[Geodesate[Polyhedron[Dodecahedron], 4]];
```

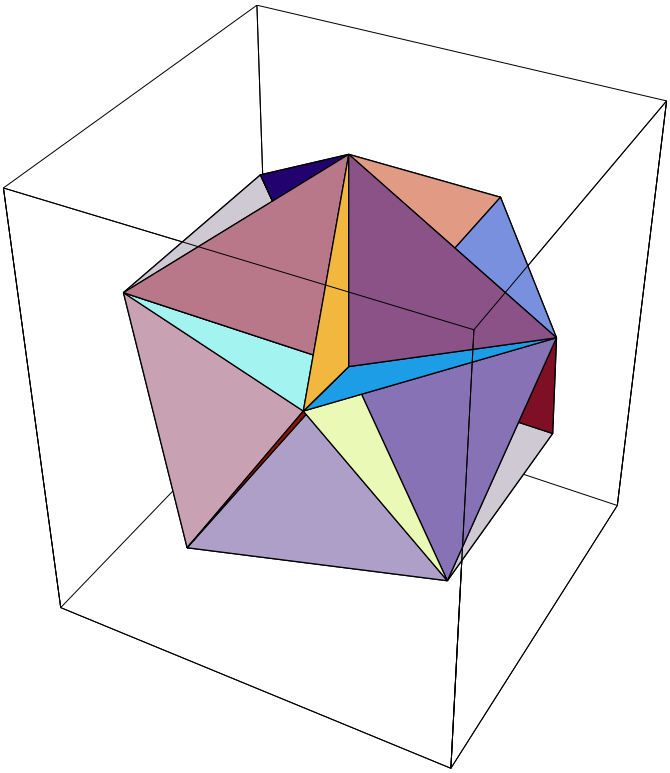


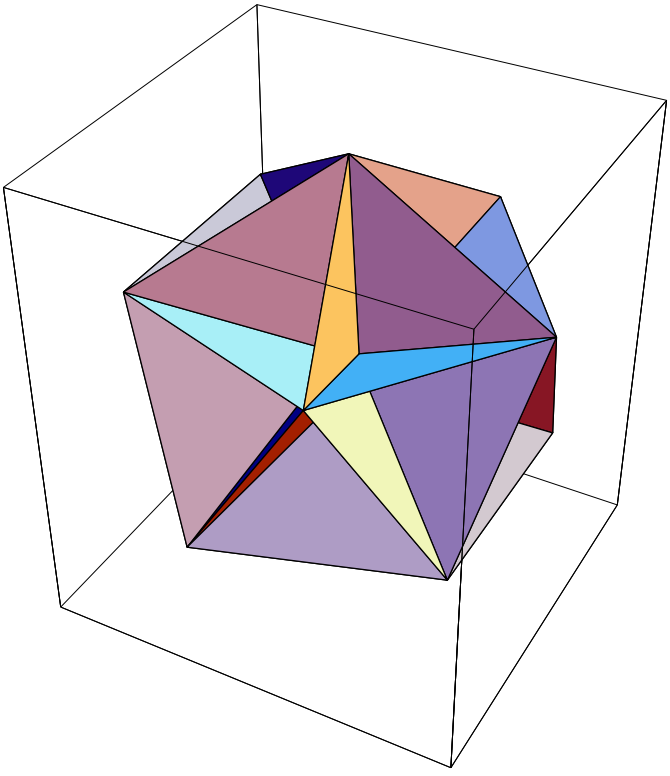
In[29]:=
```
Show[Geodesate[Polyhedron[Dodecahedron], 5]];
```

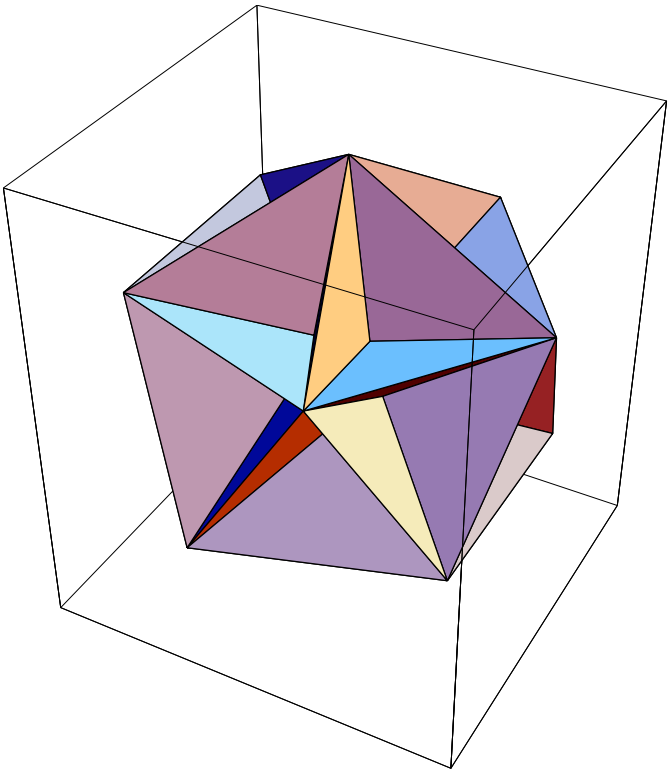


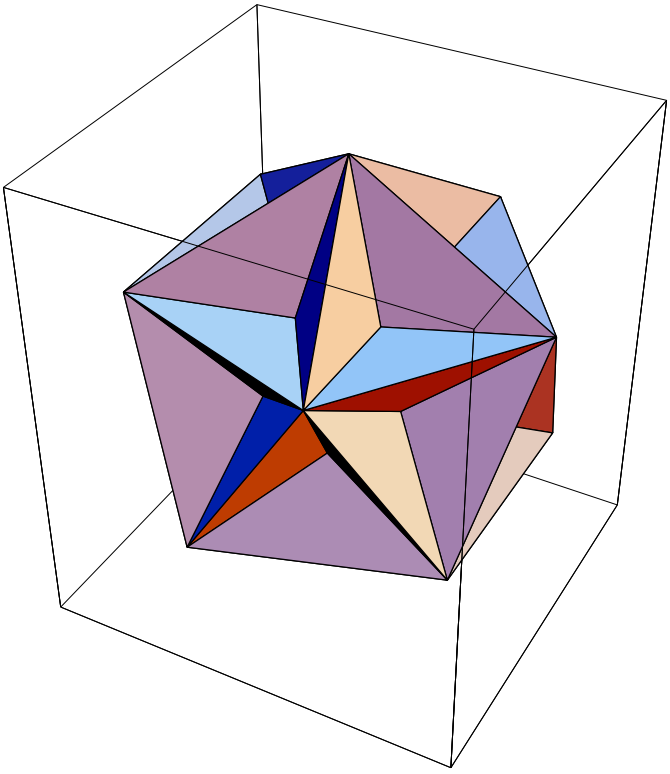
■ **Animationen: ===> Animate selected Graphics in "Cell"  !!!!**

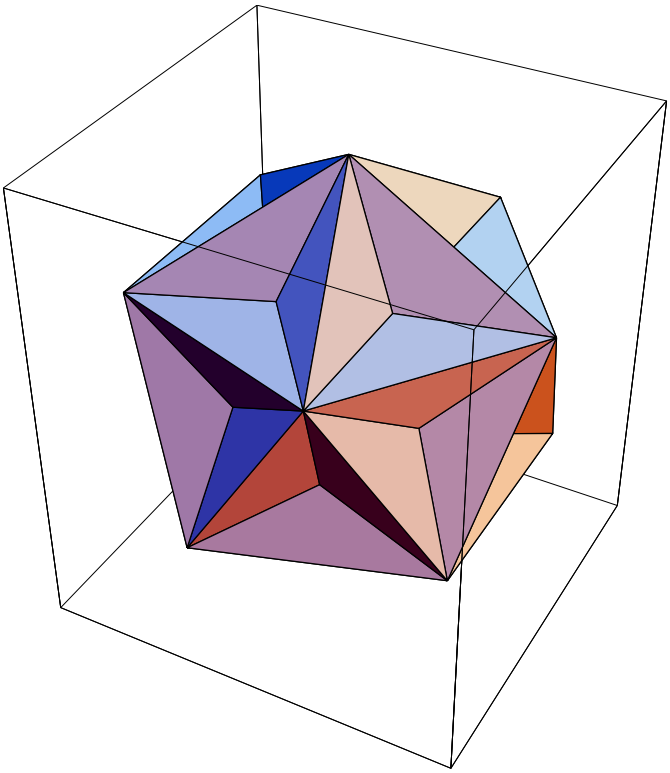
In[30]:=
```
Table[Show[Stellate[Polyhedron[Icosahedron], 0.2 k]], {k, 0, 30}];
```

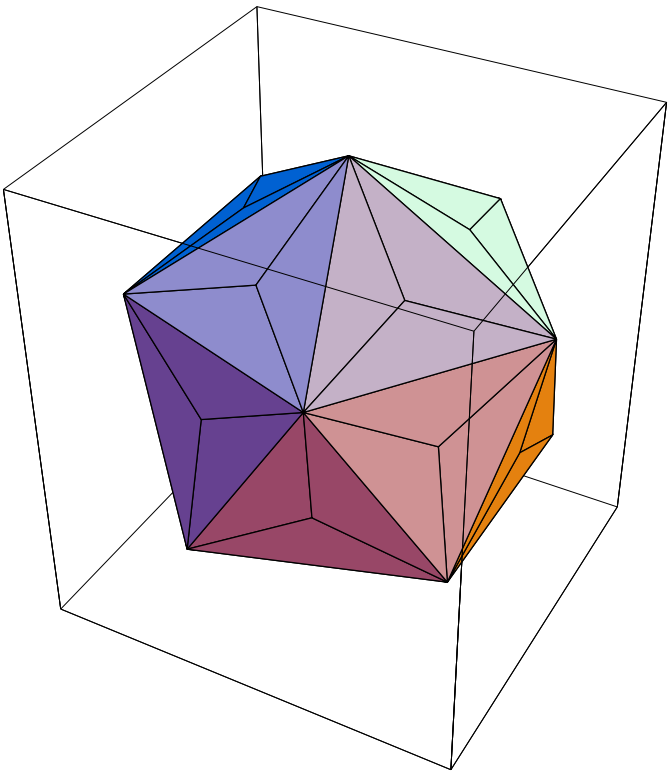


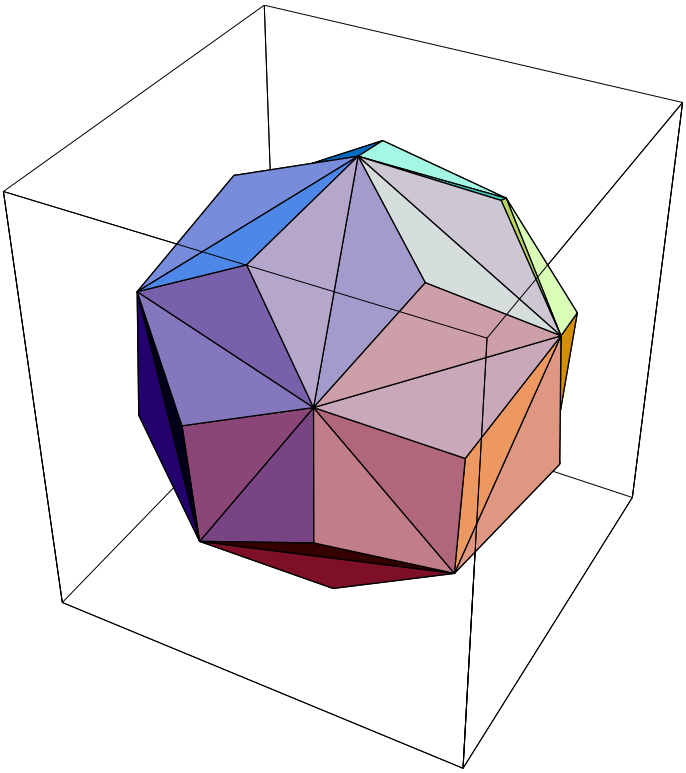


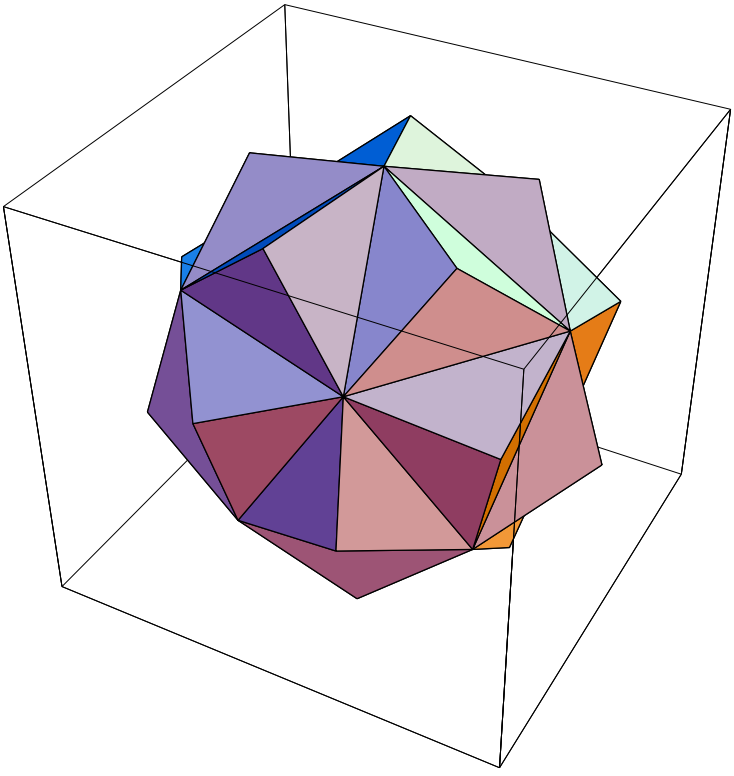


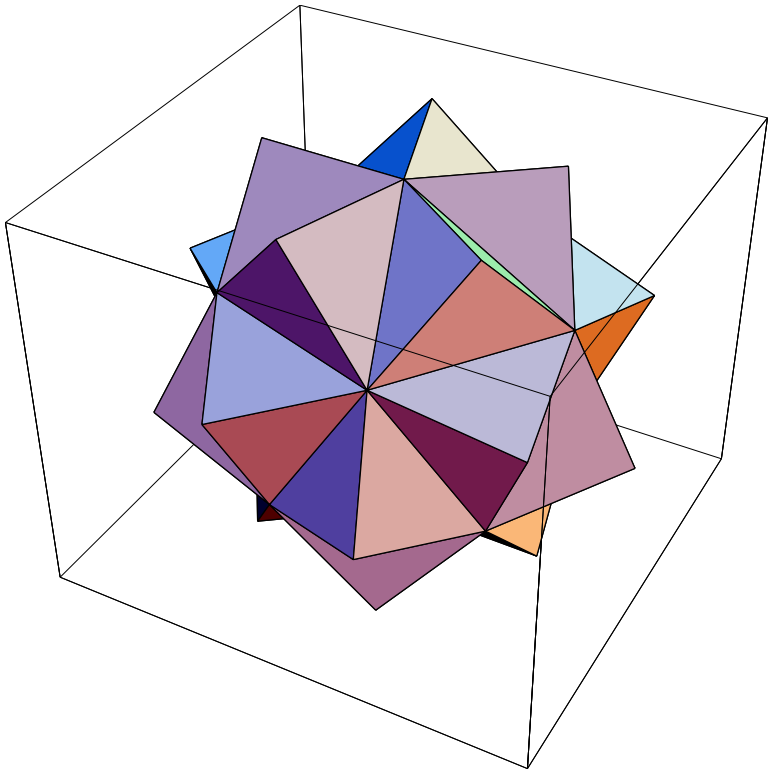


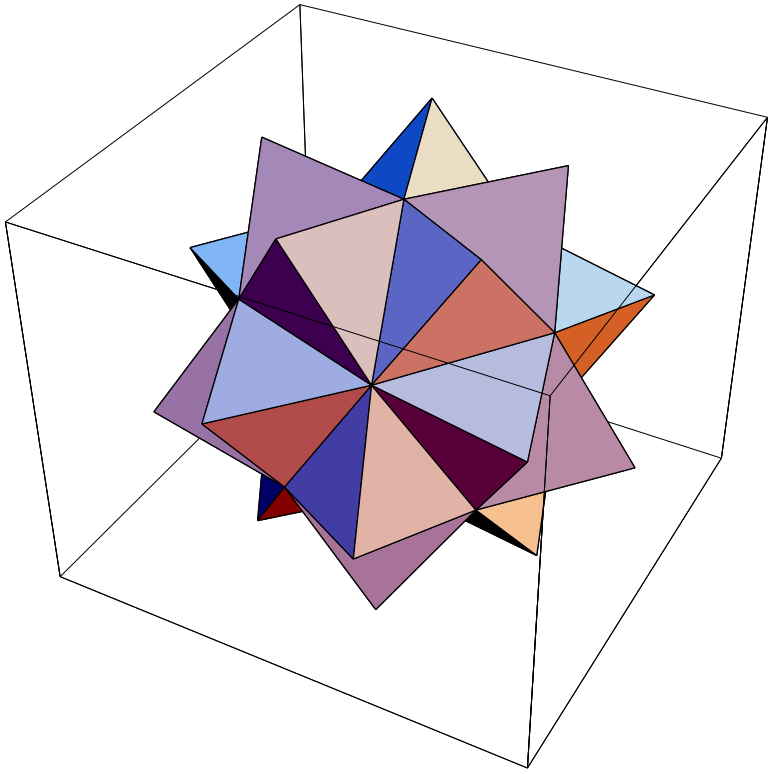


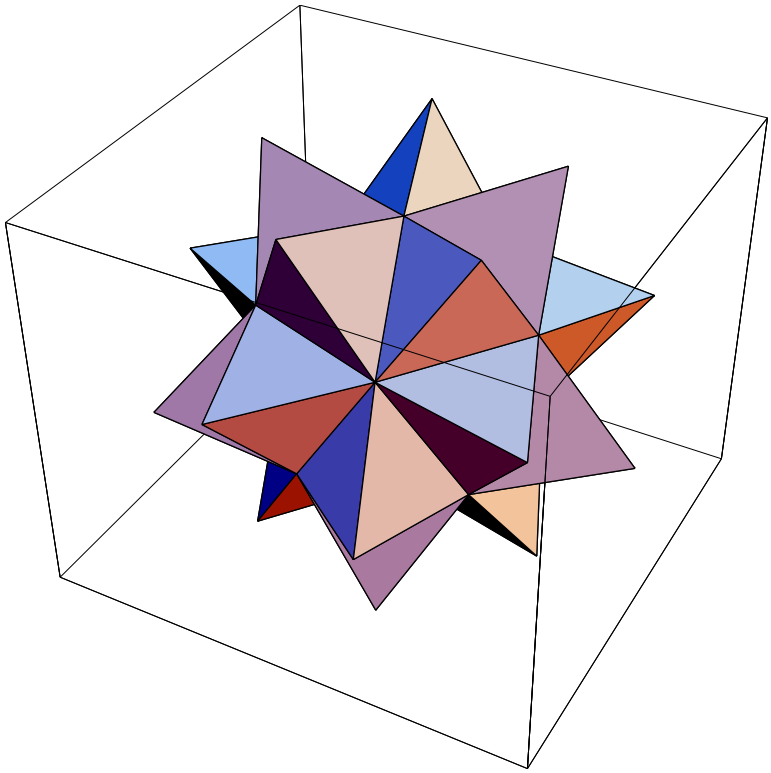


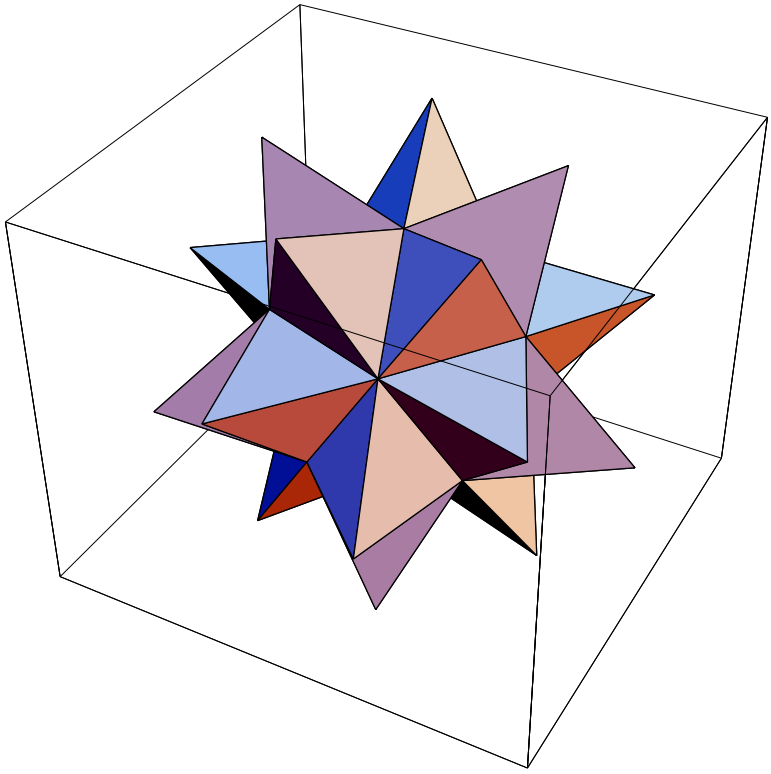


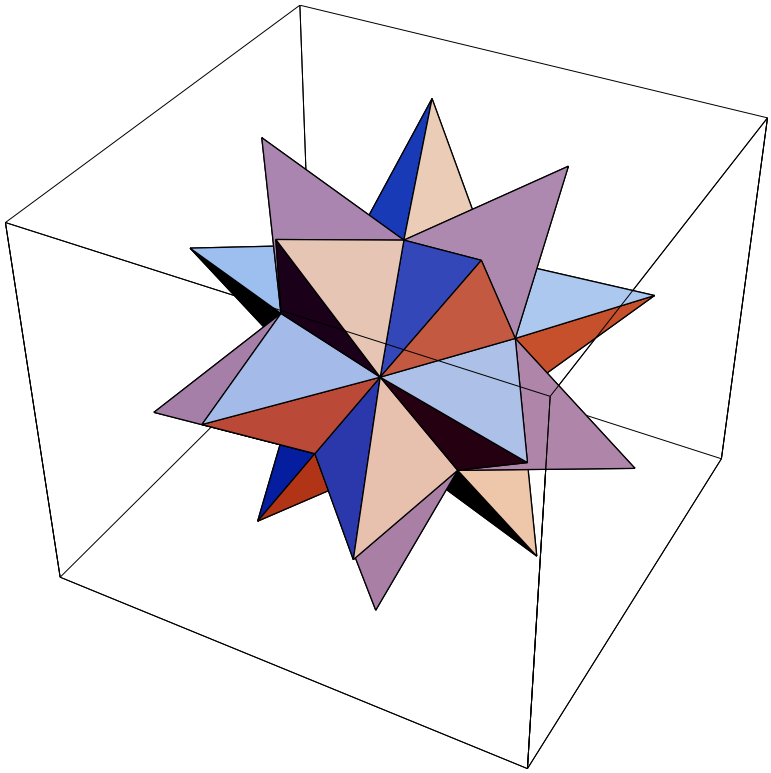


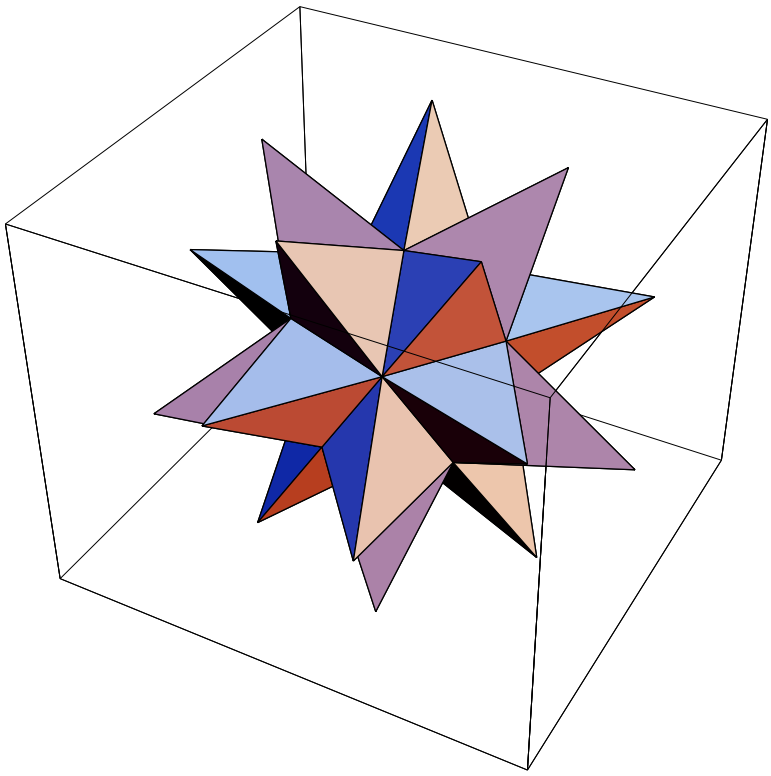


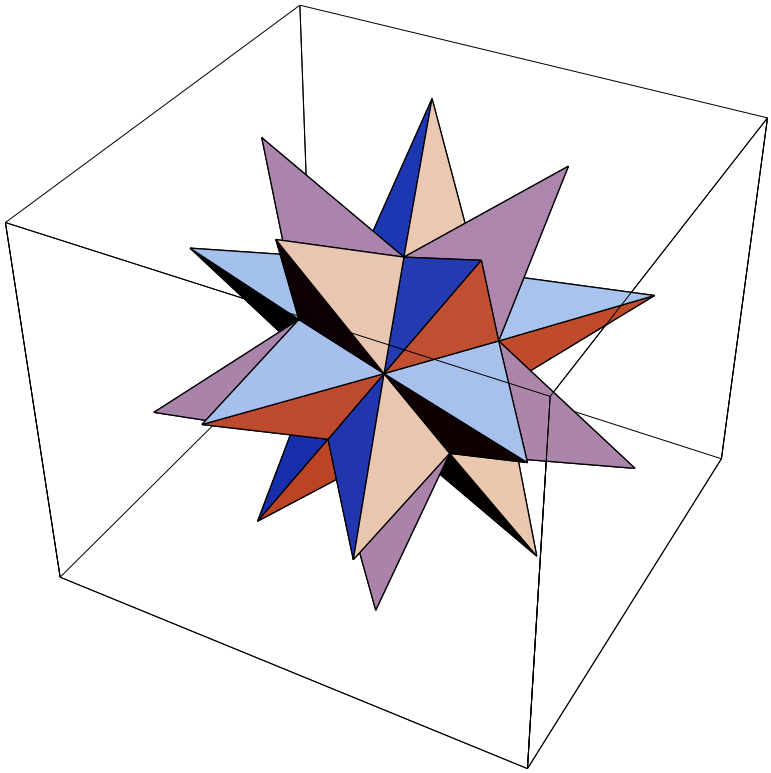


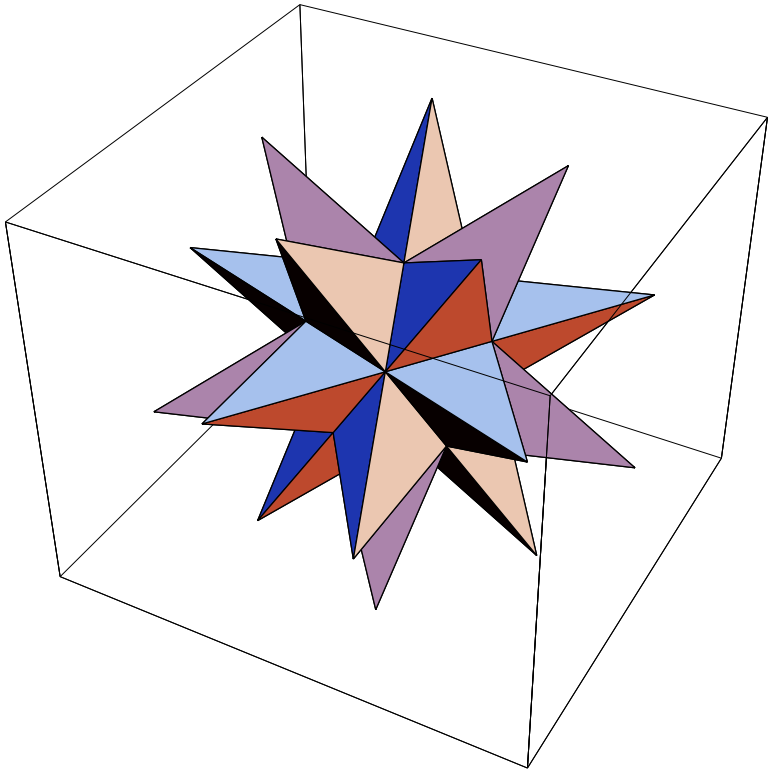


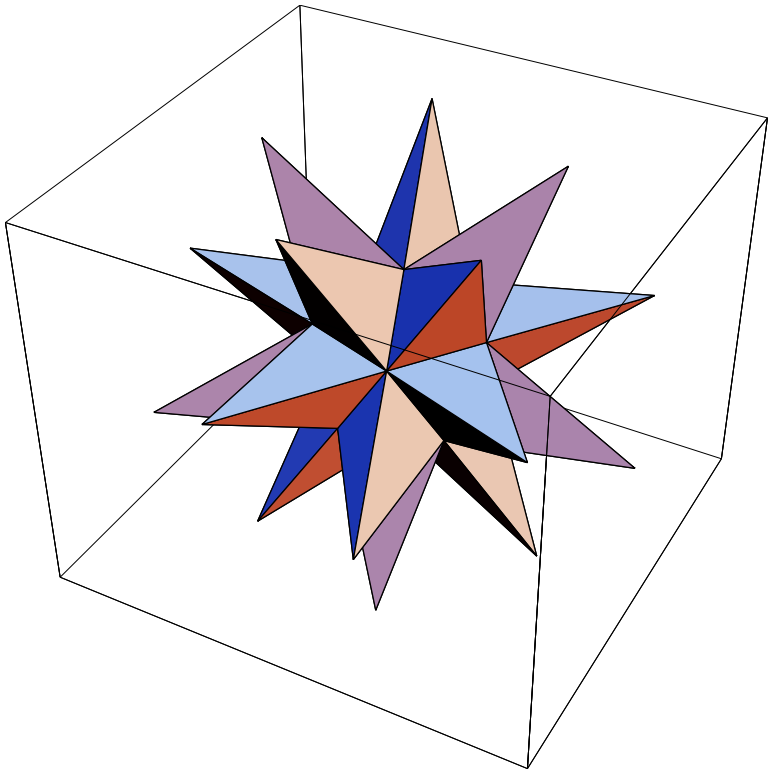


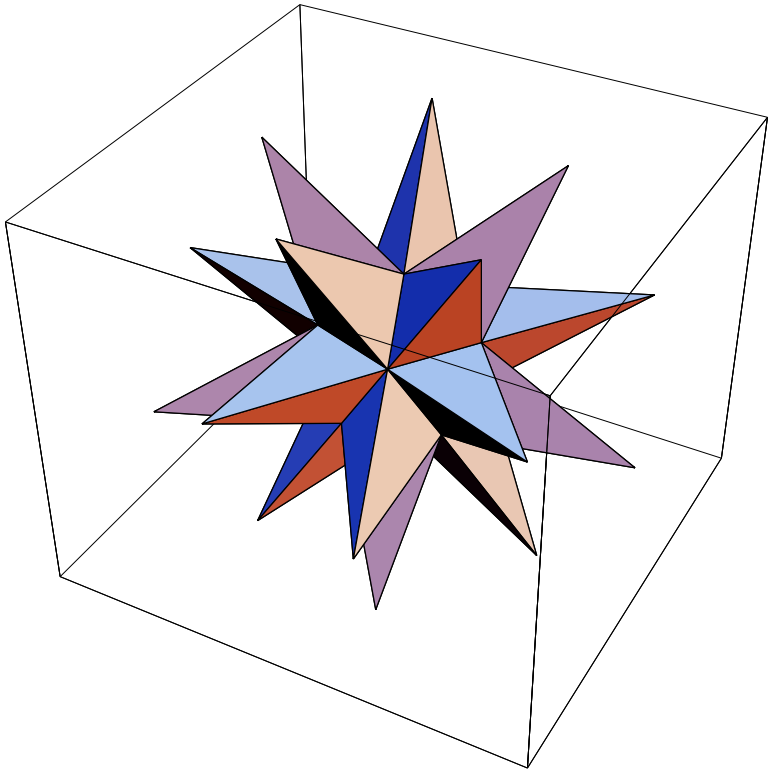


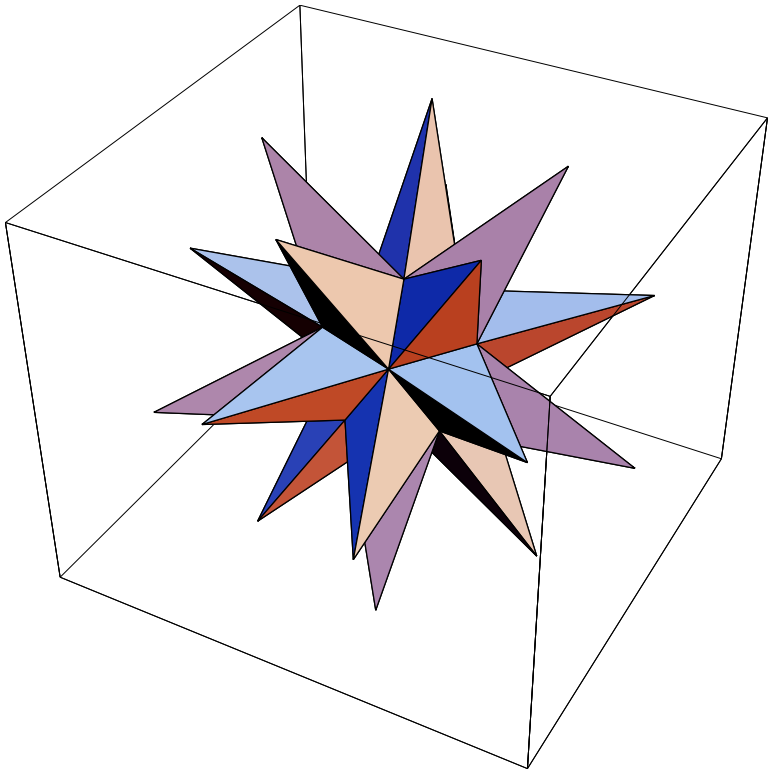


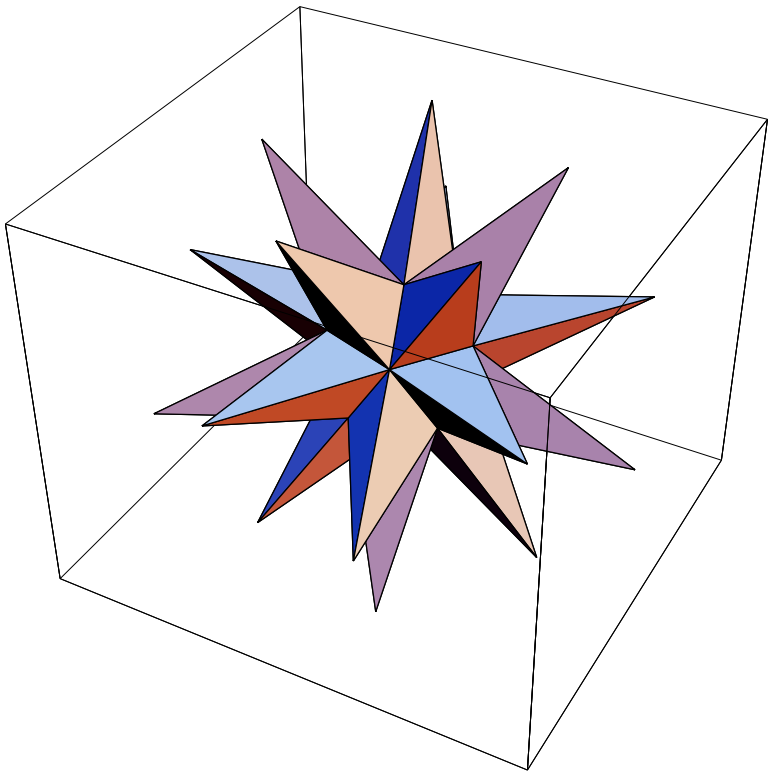


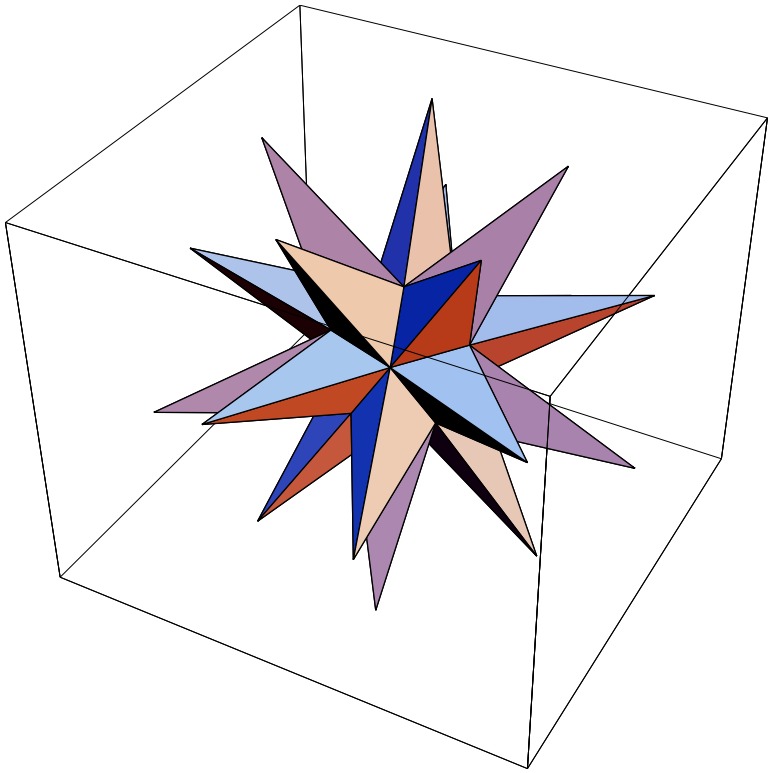


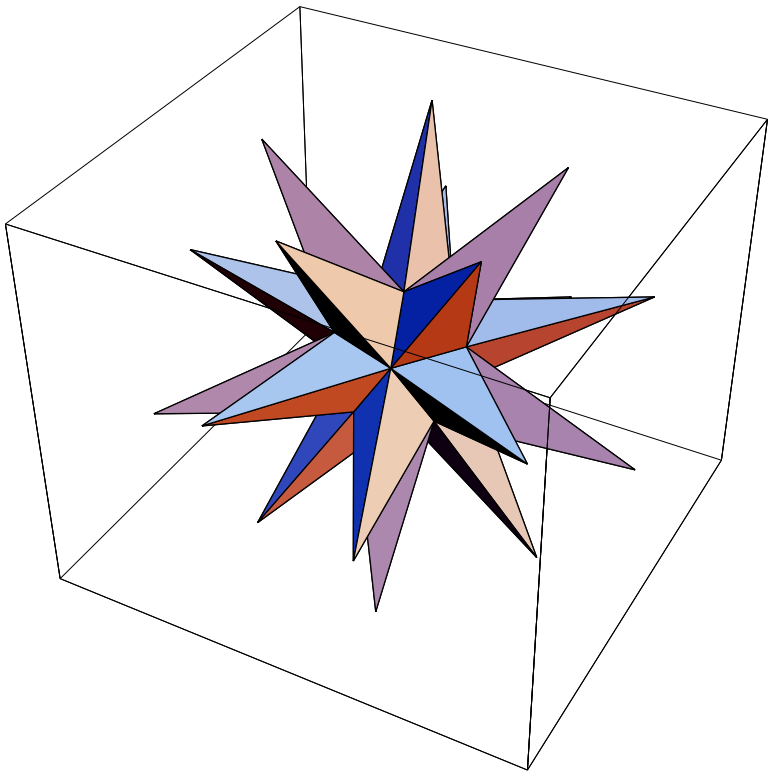


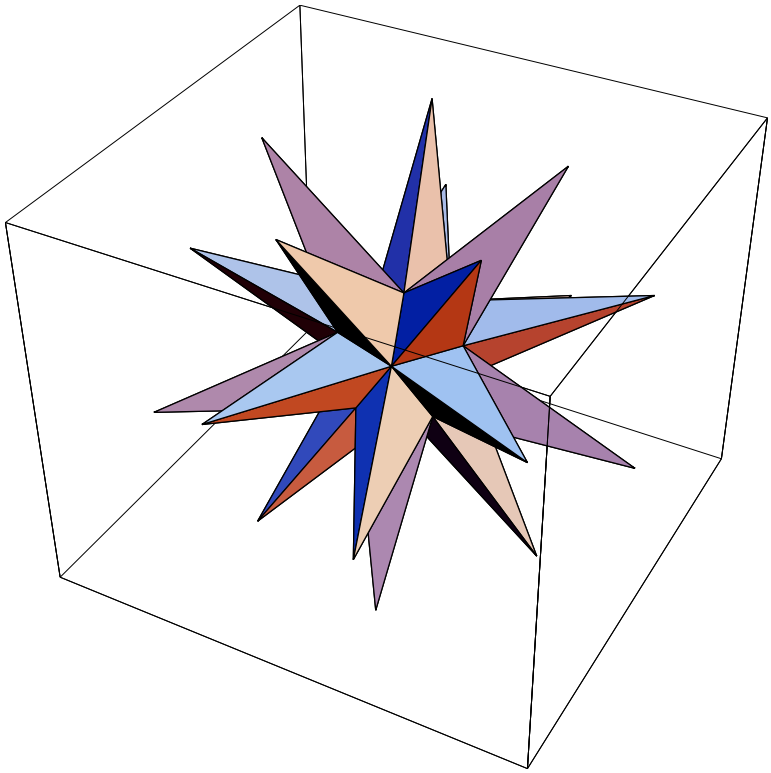


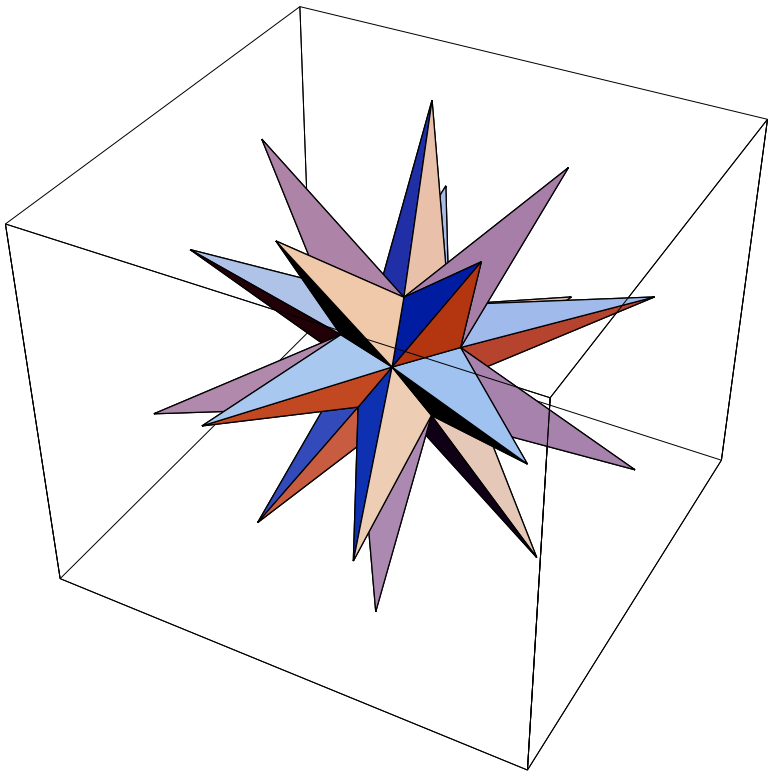


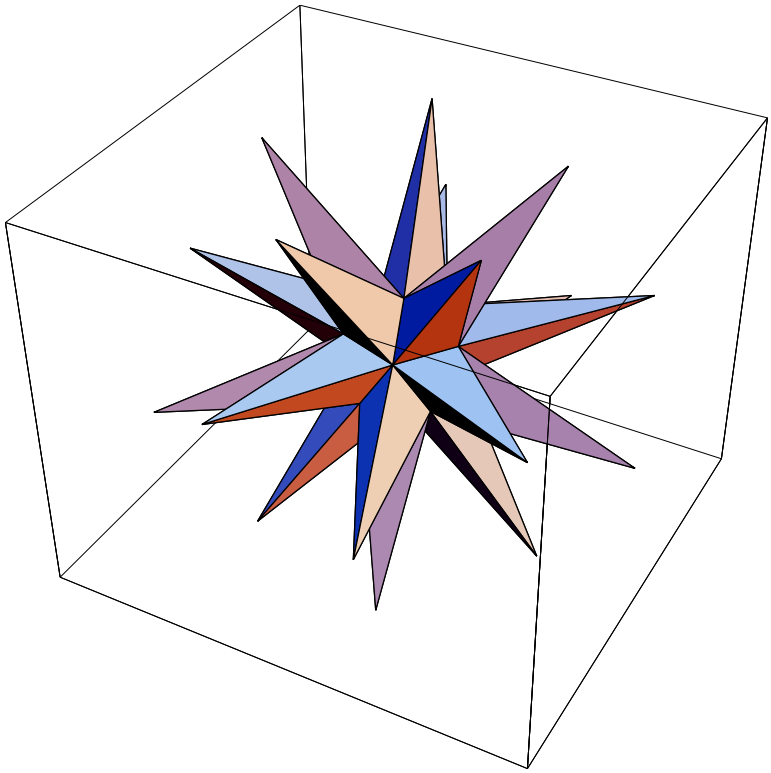


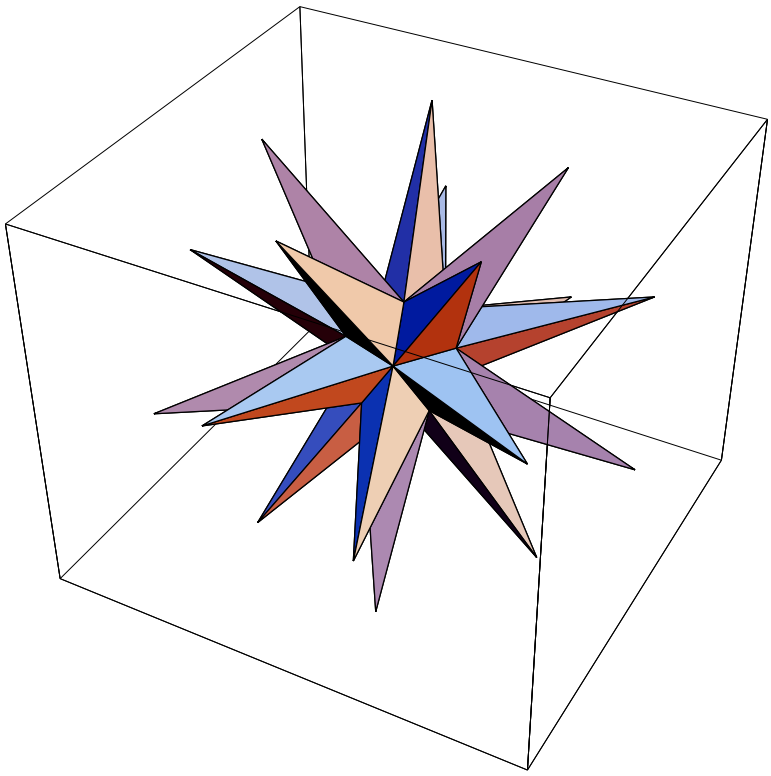


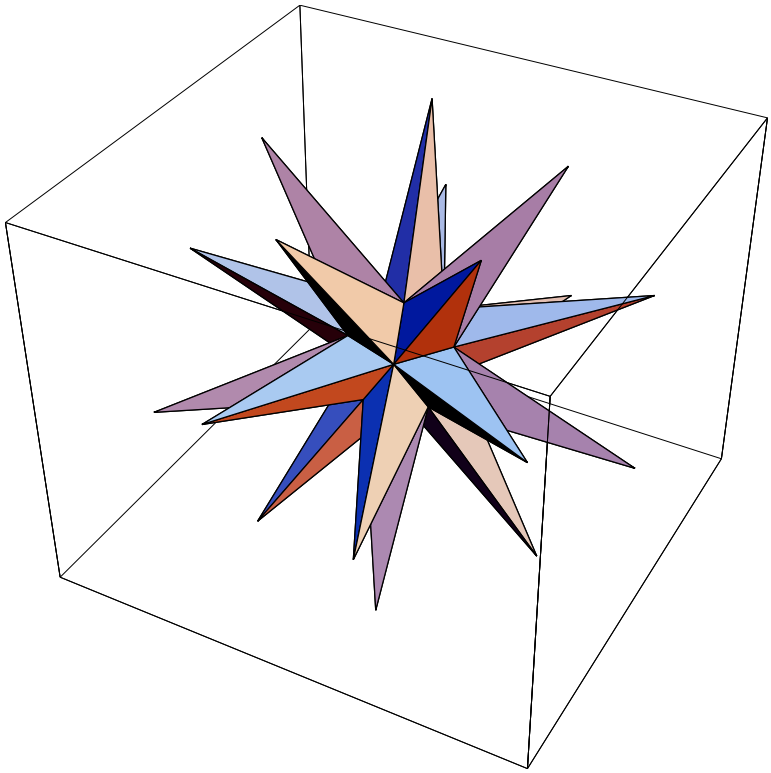


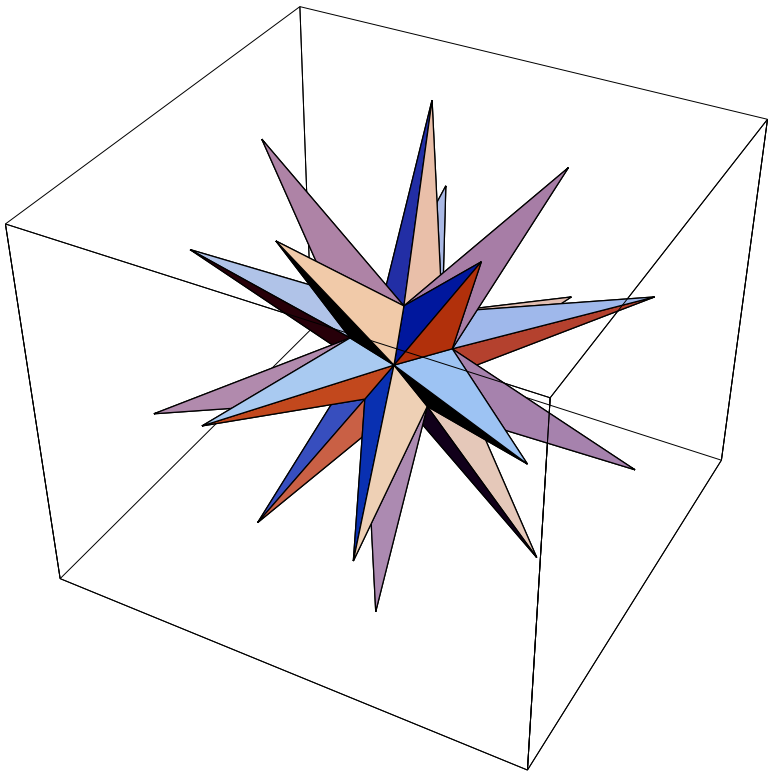


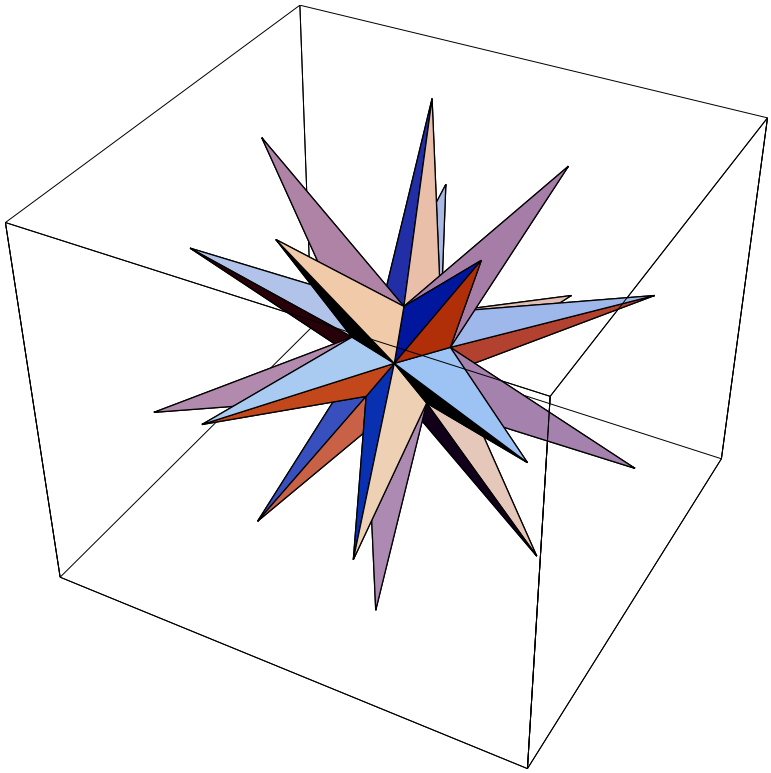


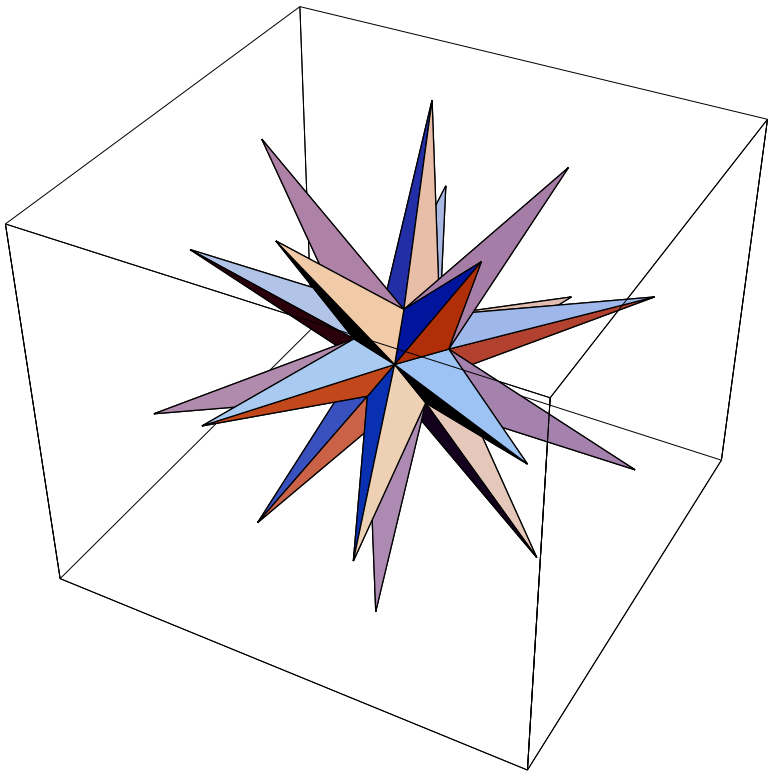


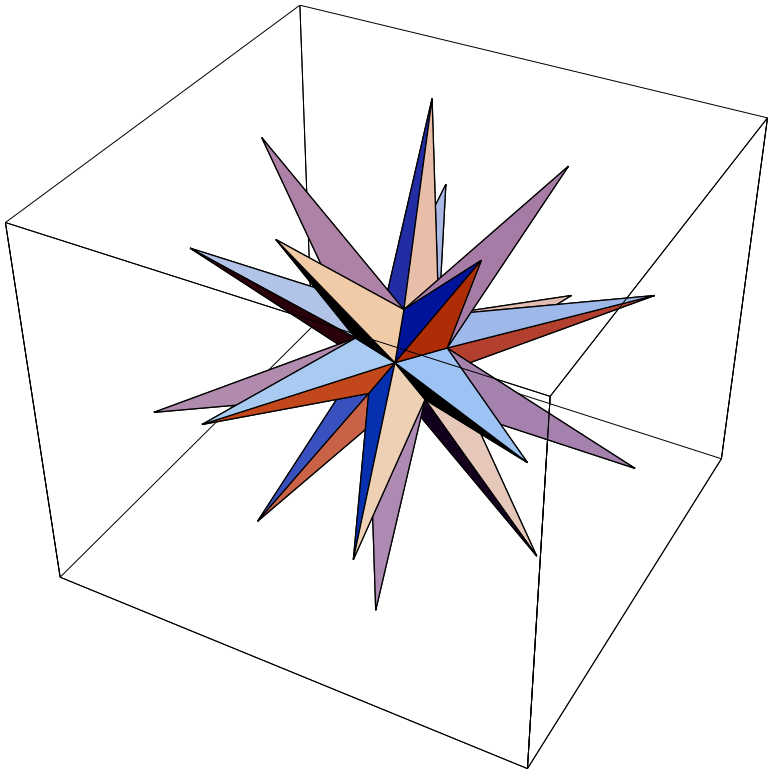




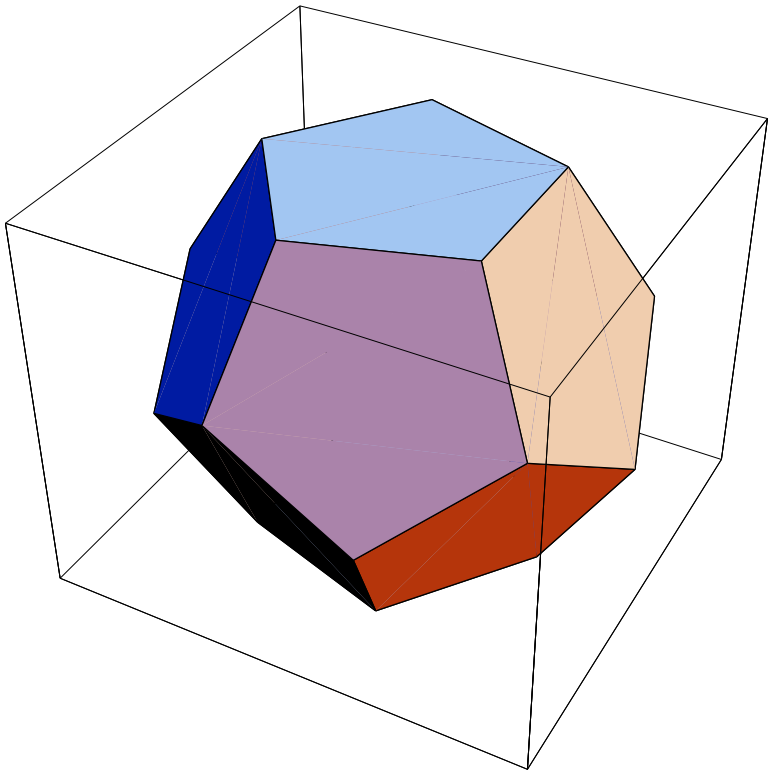


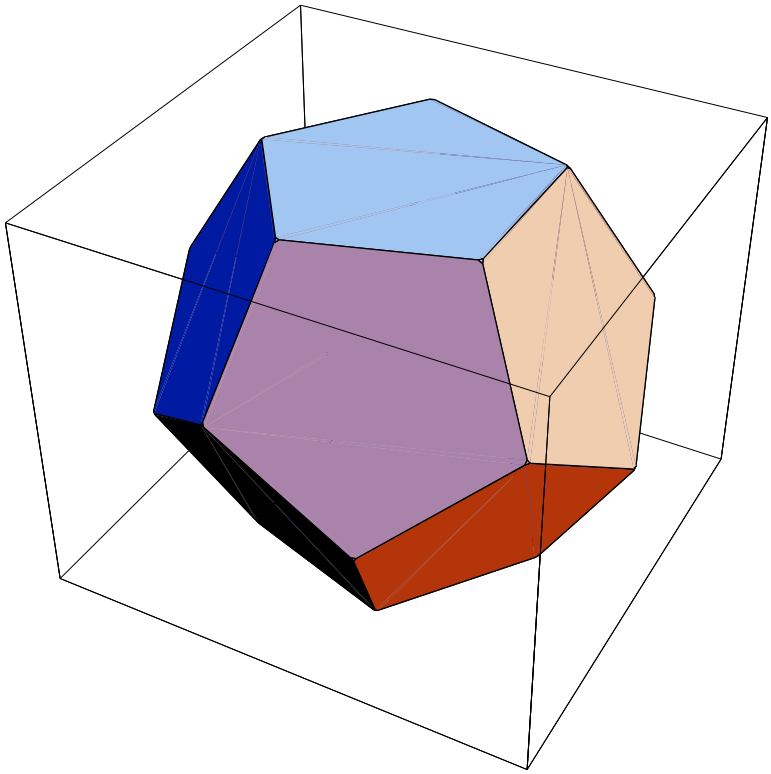


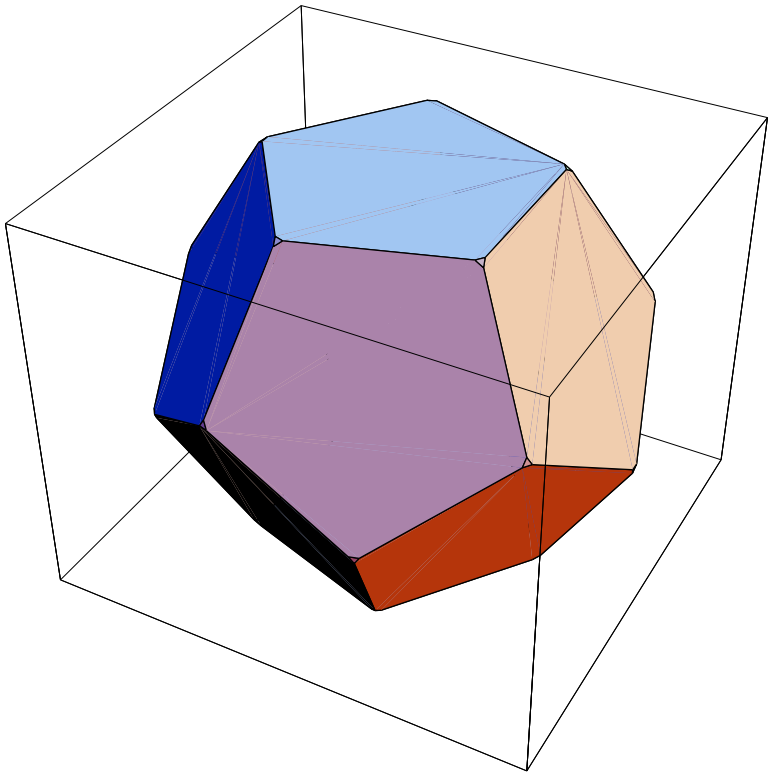


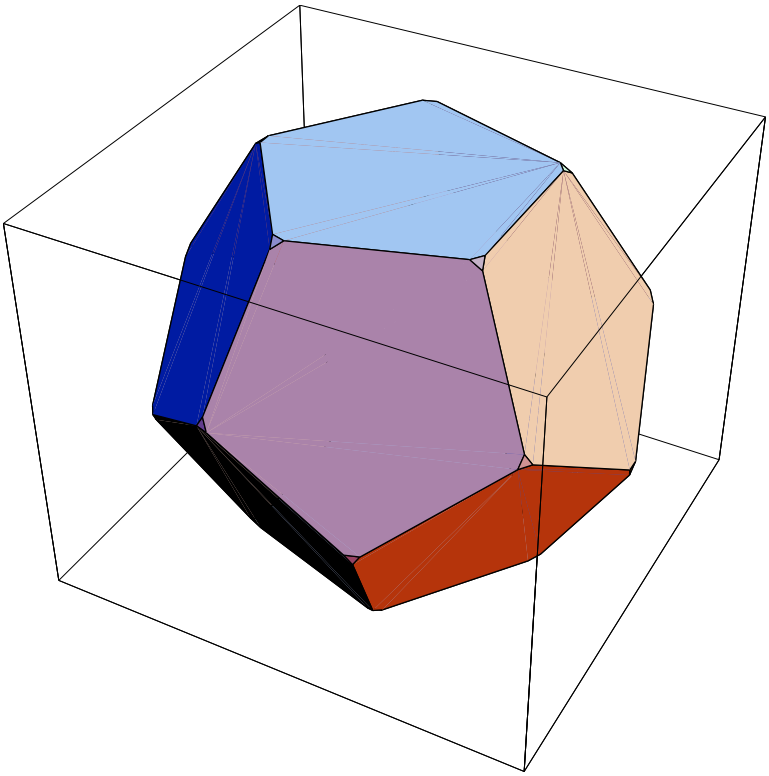


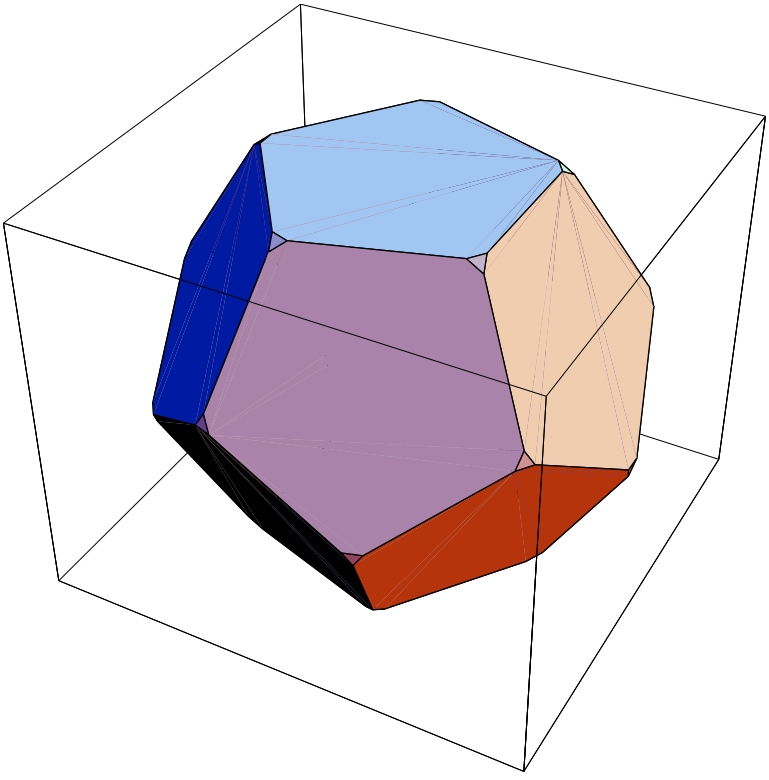
In[285]:=
```
Table[Show[Truncate[Polyhedron[Dodecahedron], 0.5 / 30 k]], {k, 0, 30}];
```

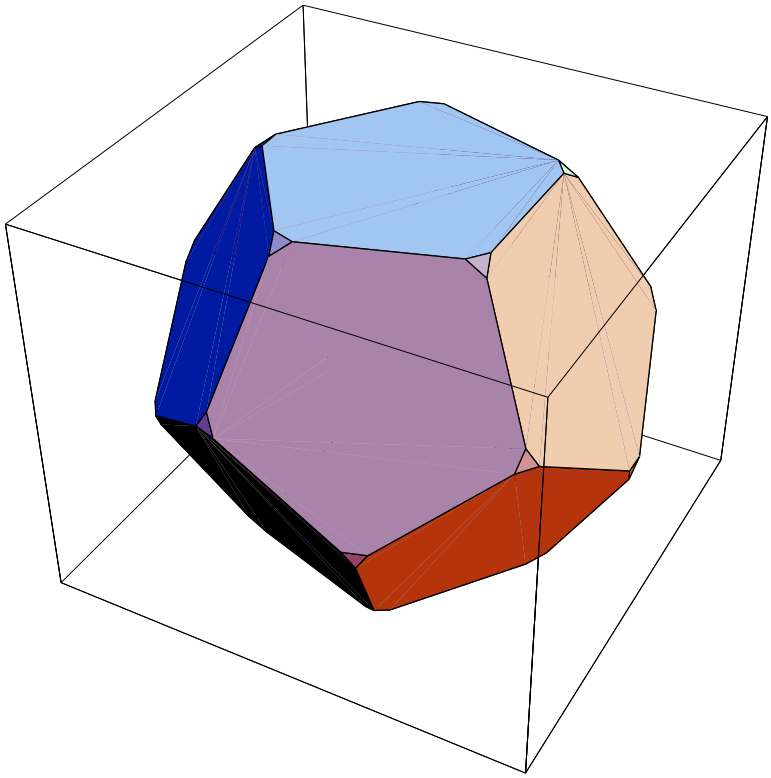


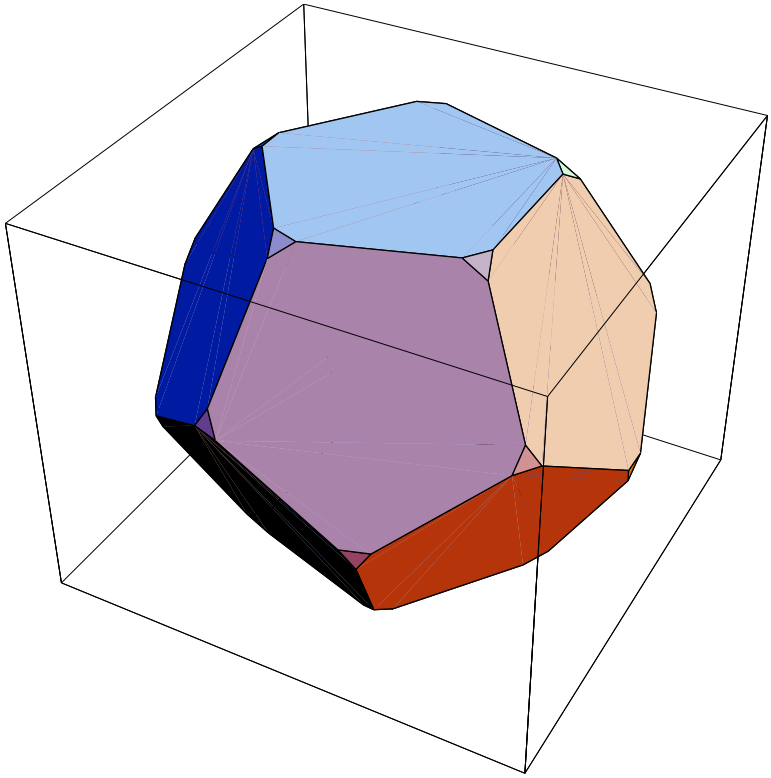


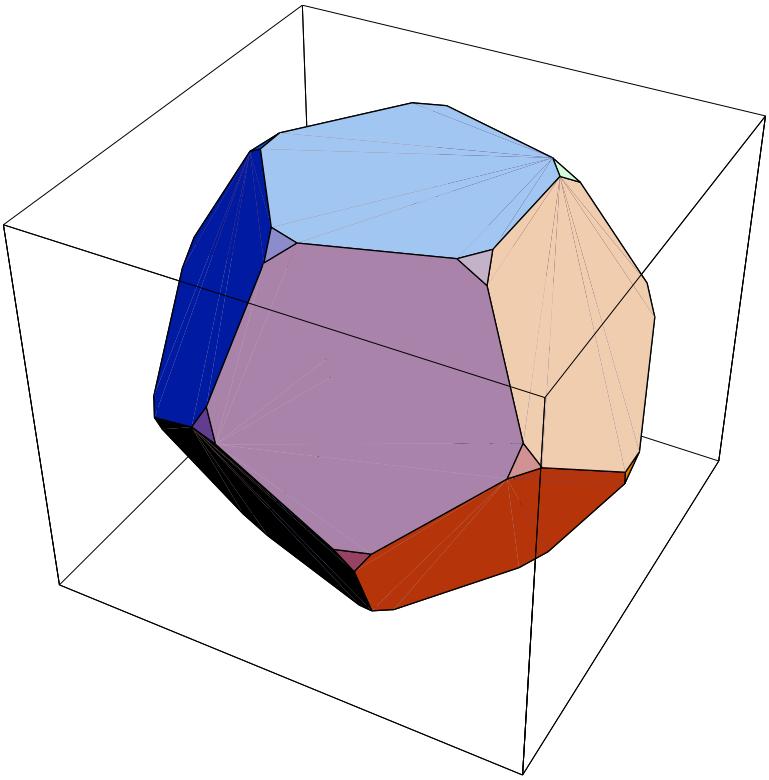


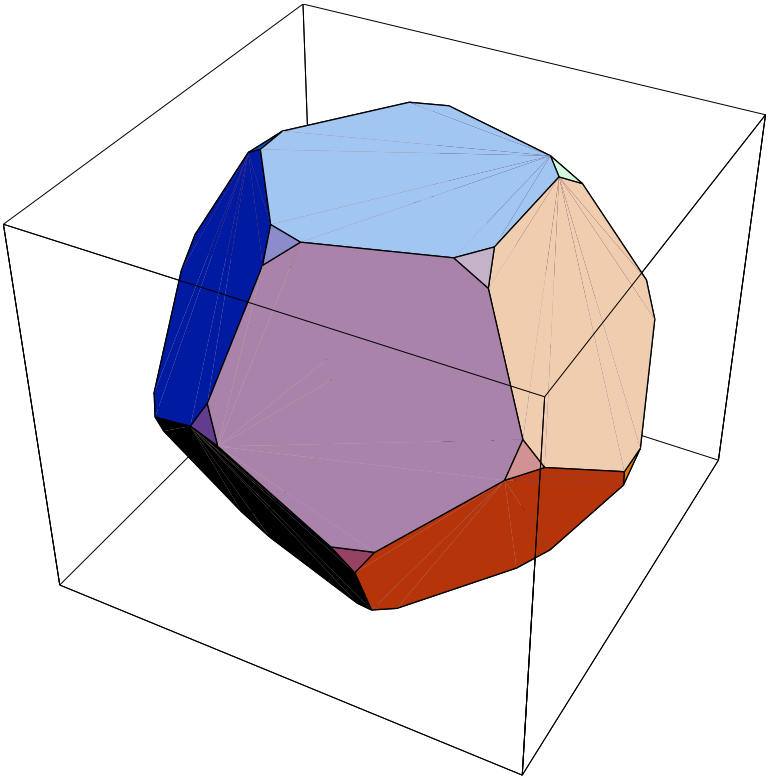


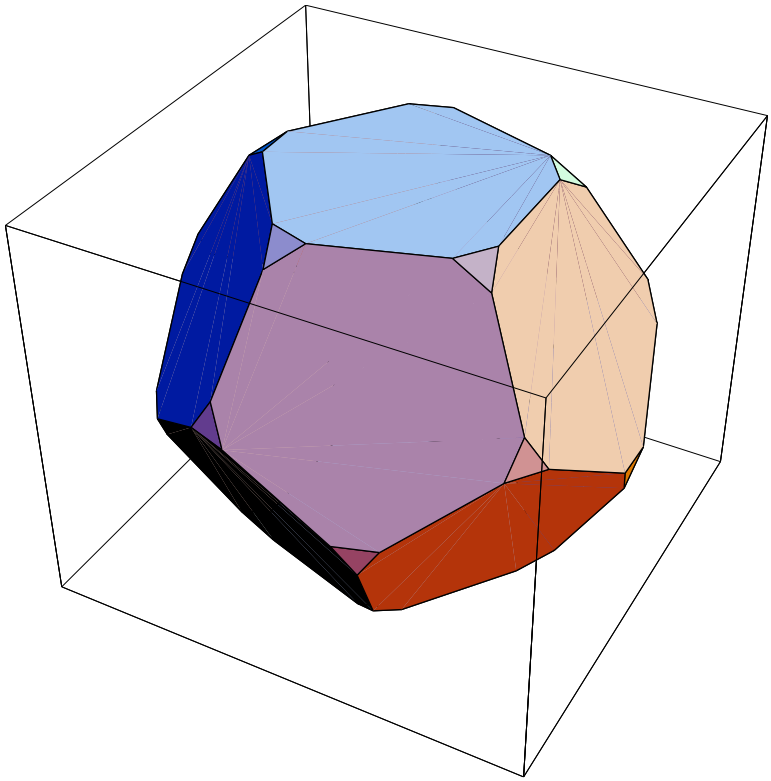


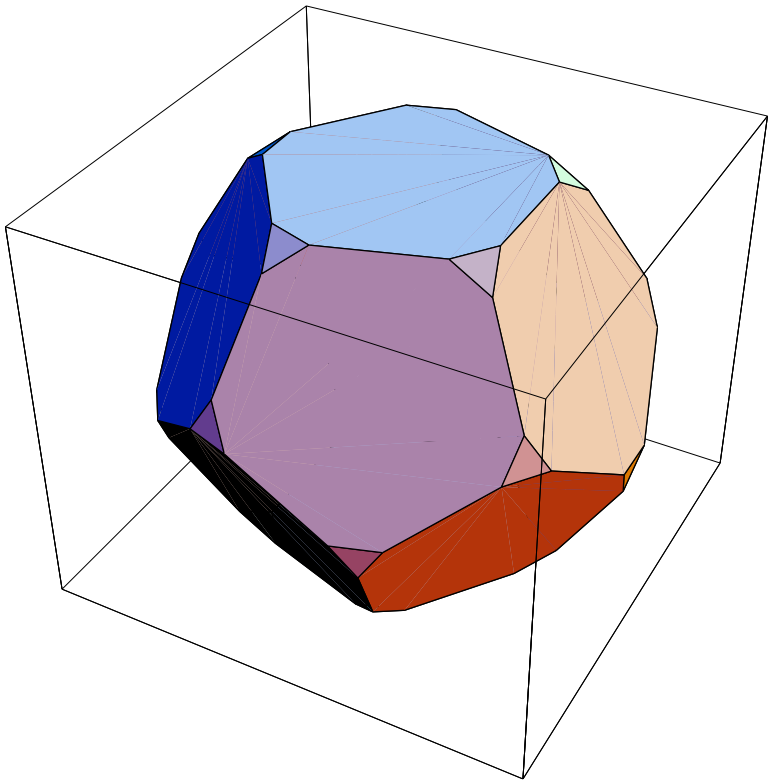


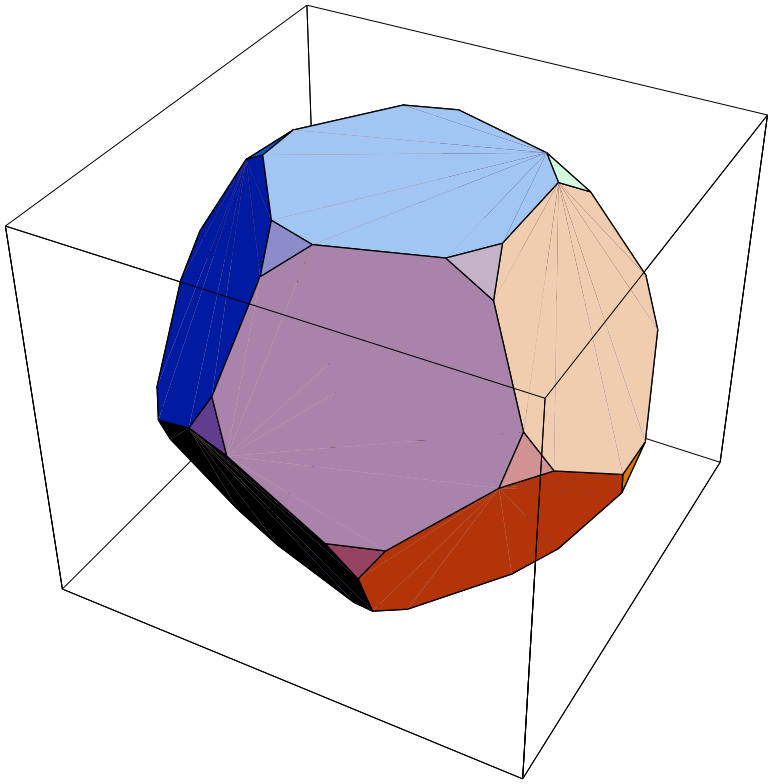


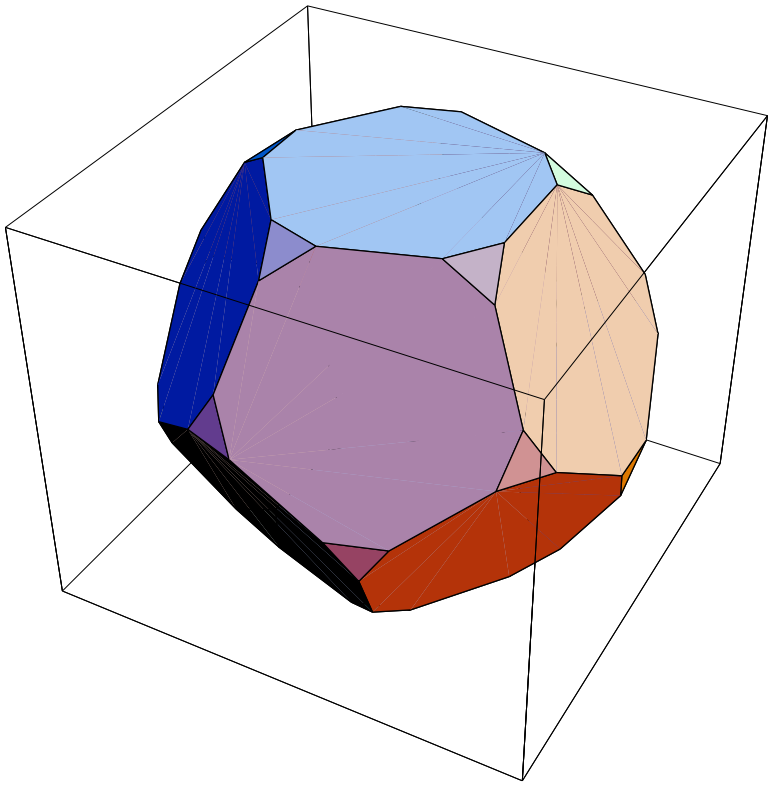


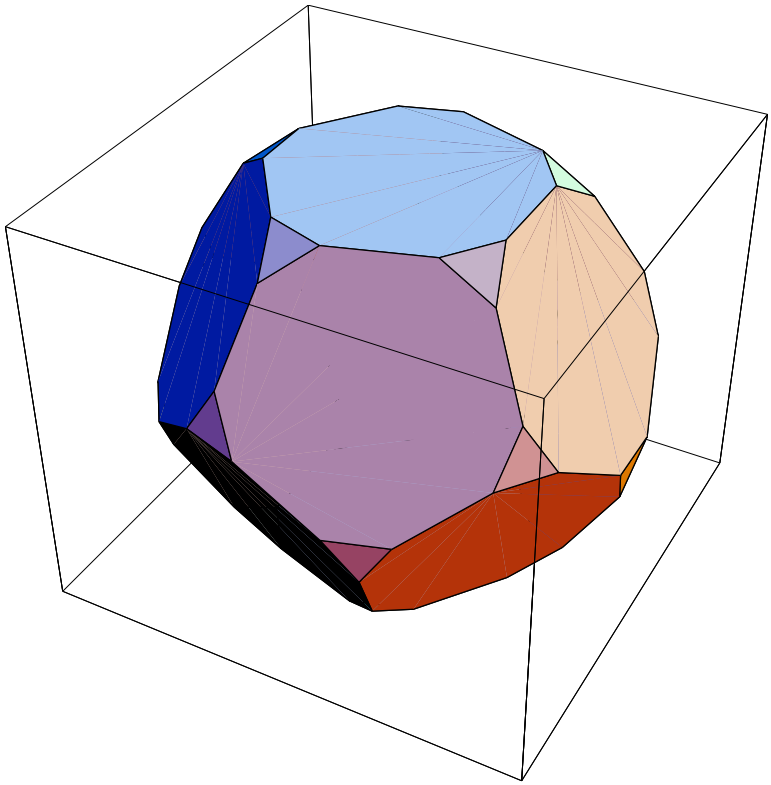


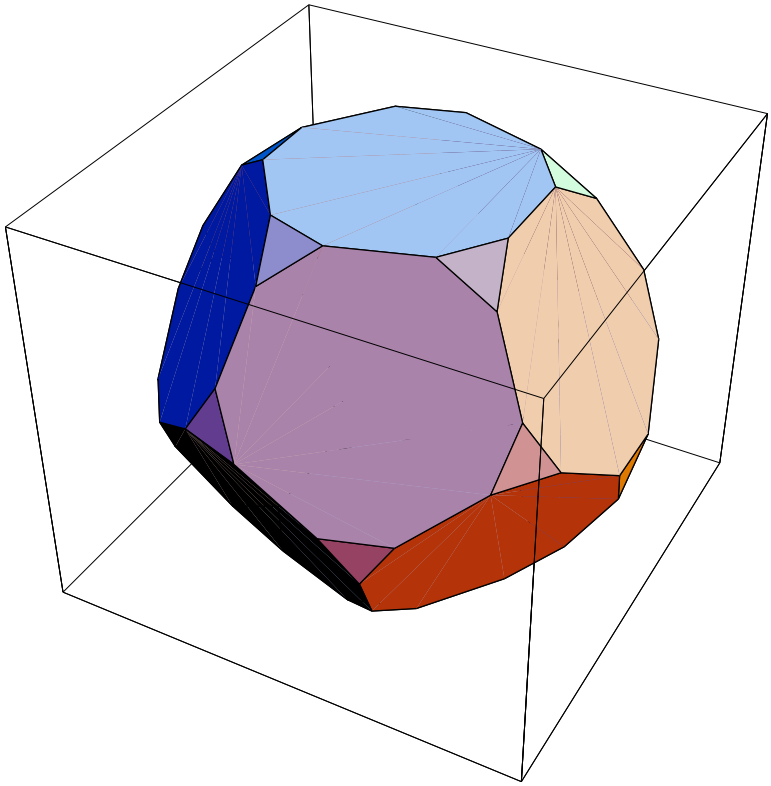


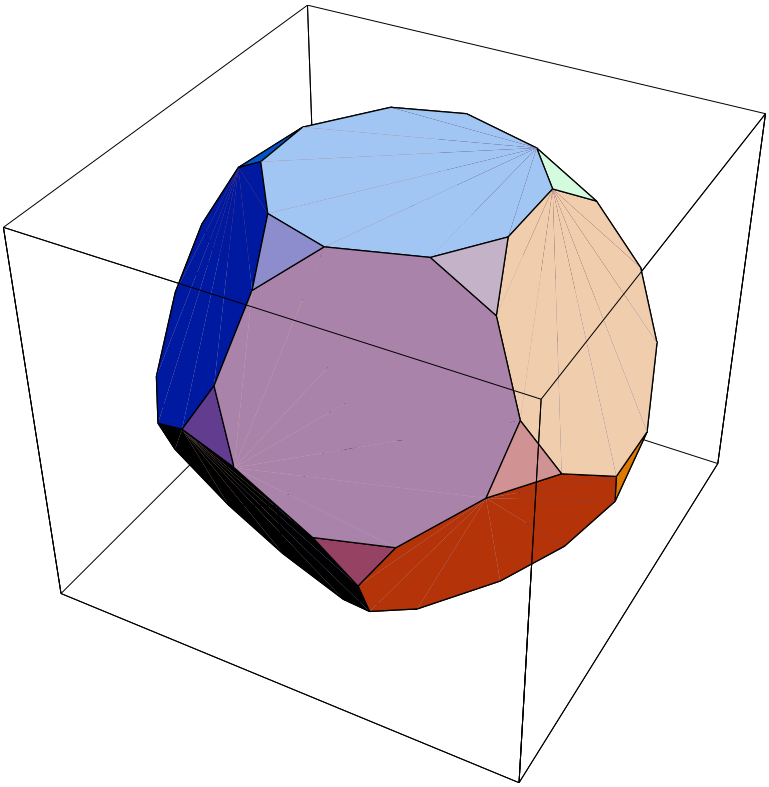


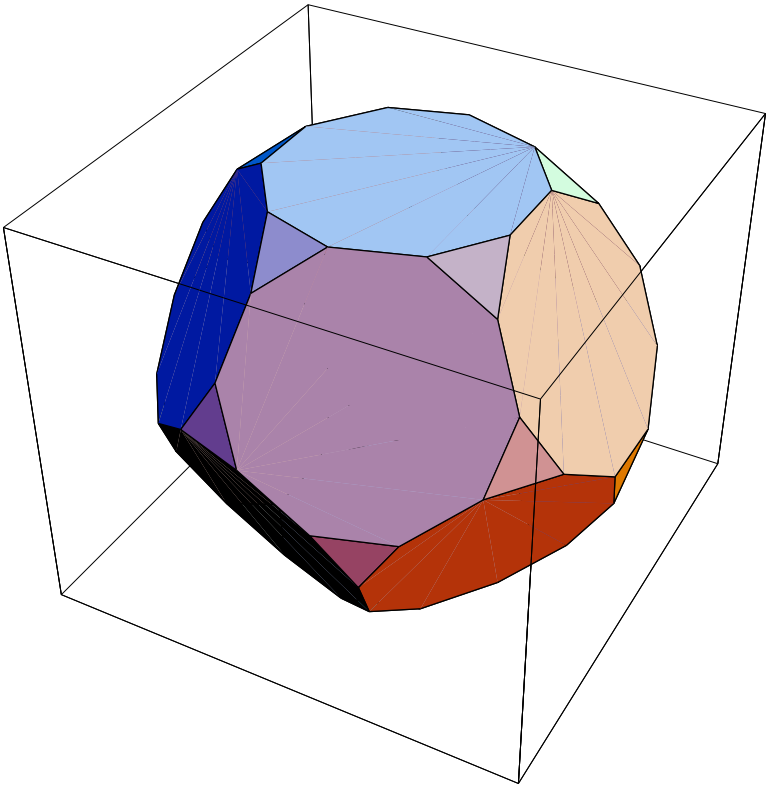


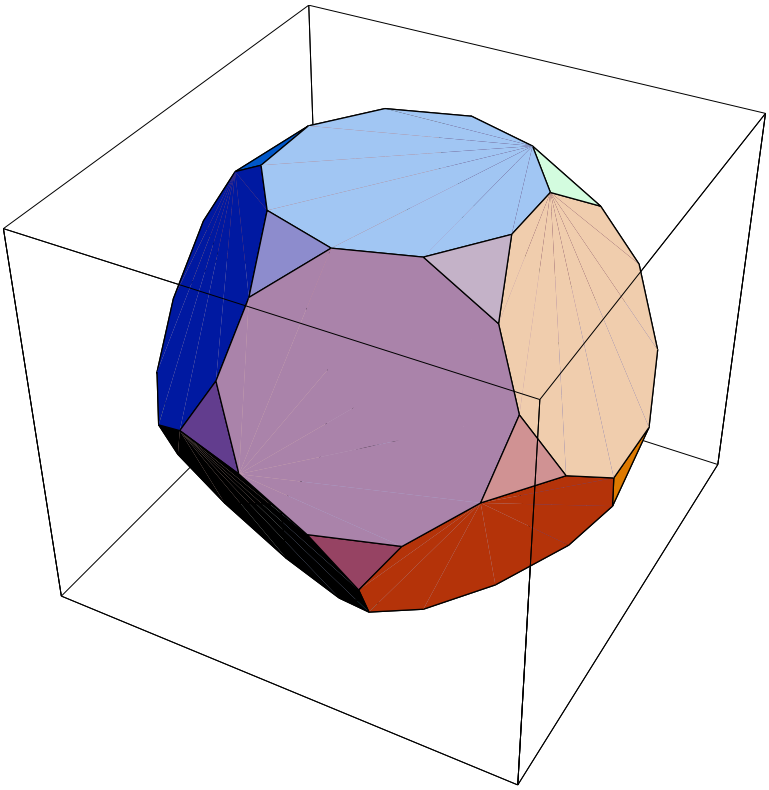


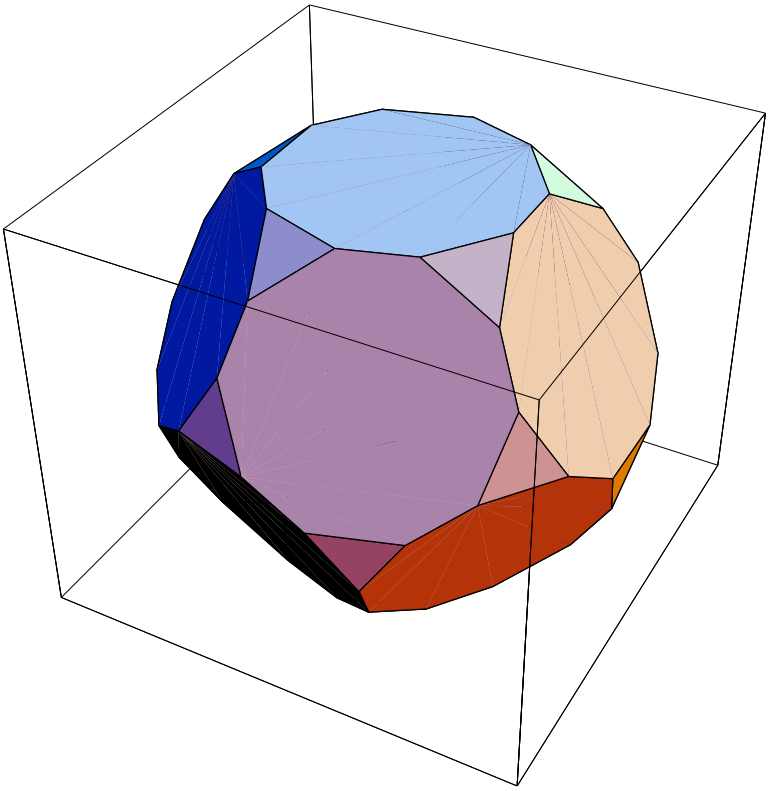


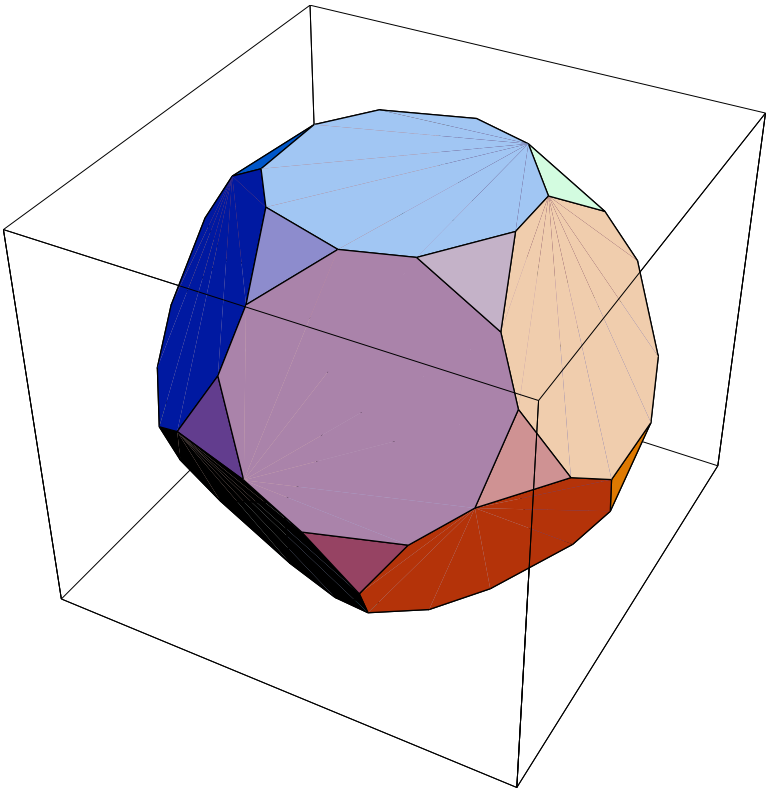


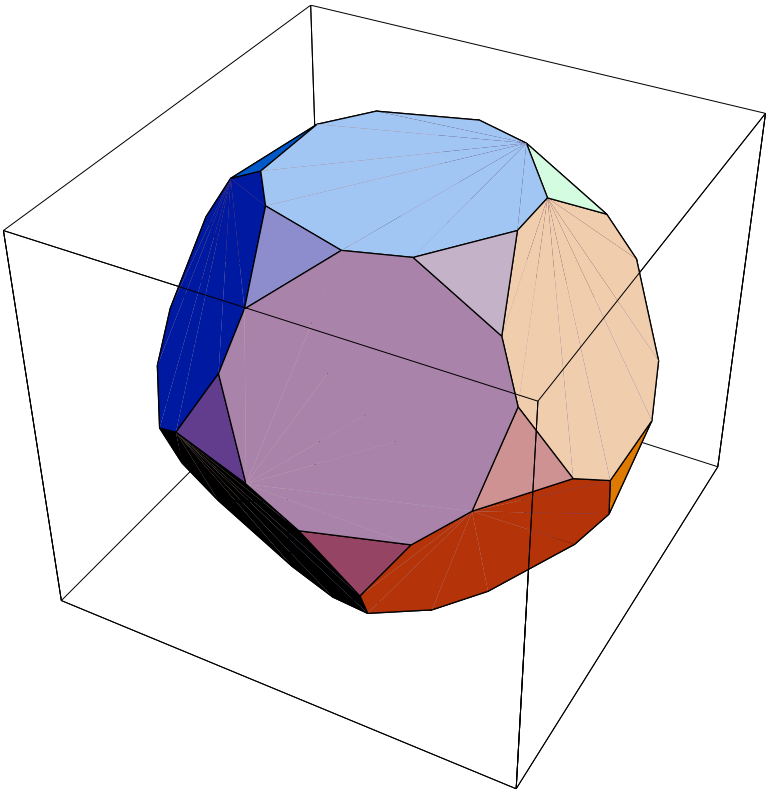


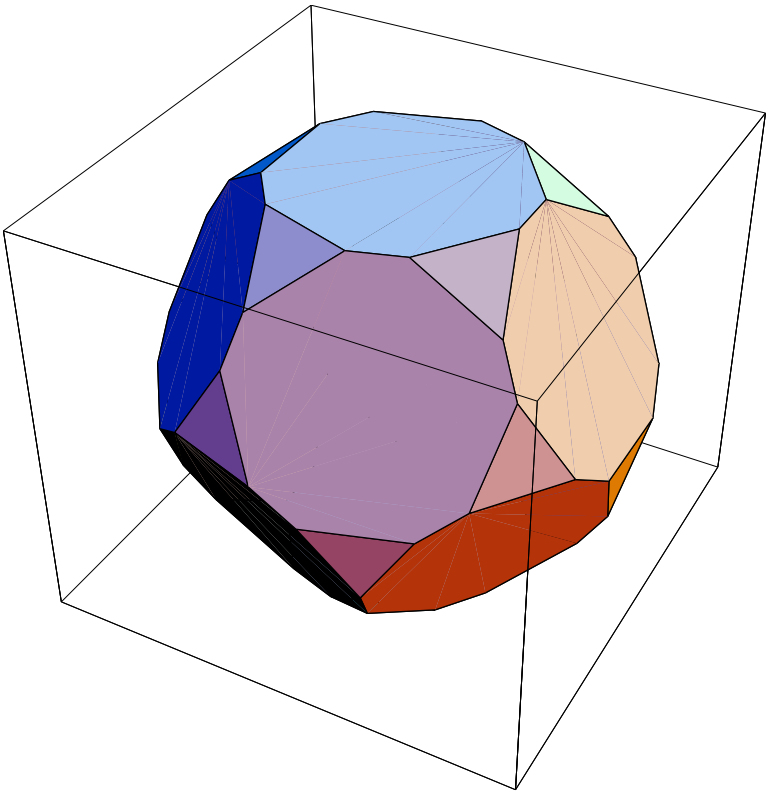


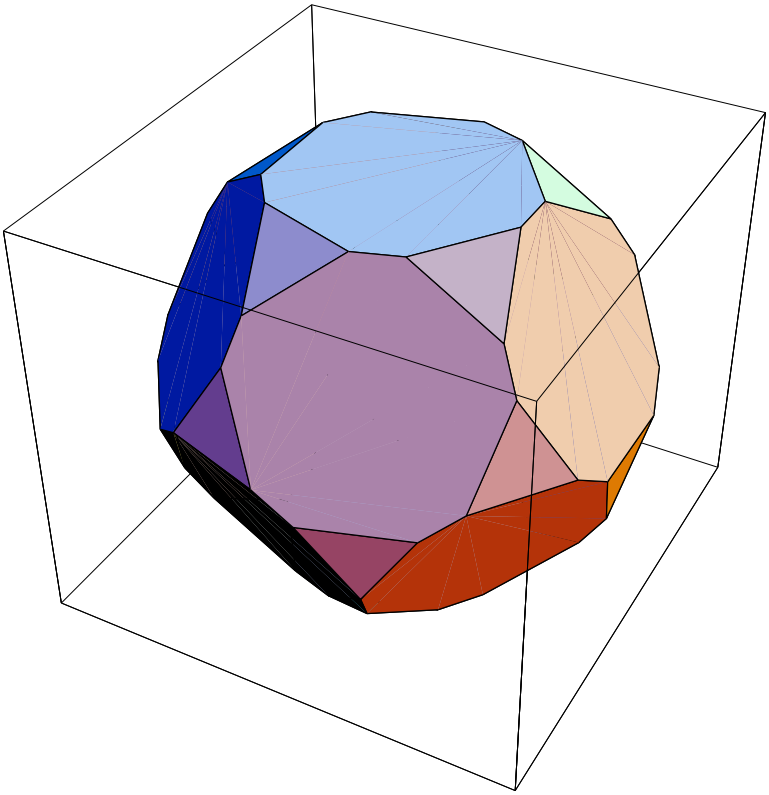


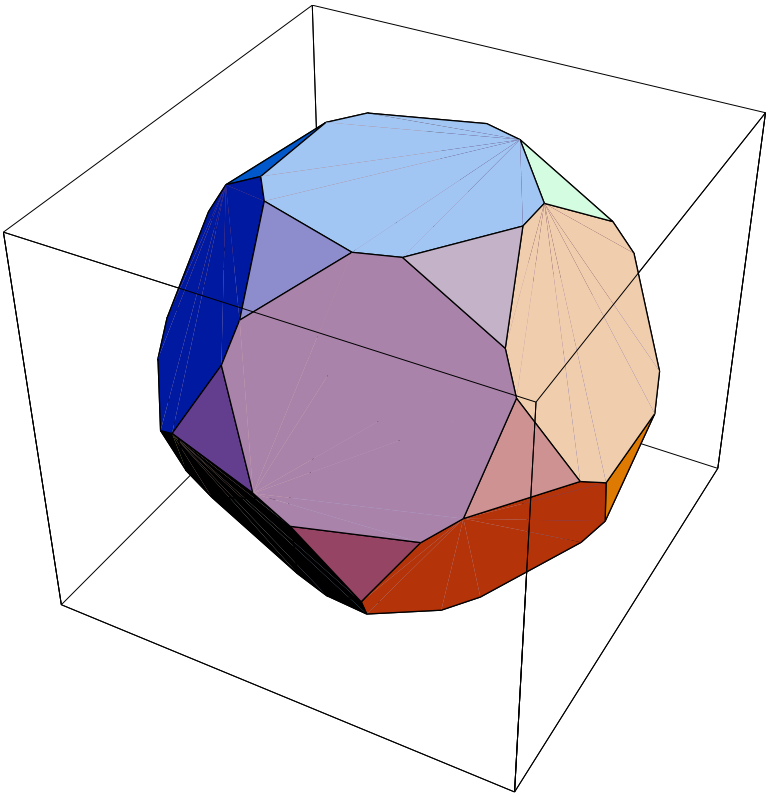


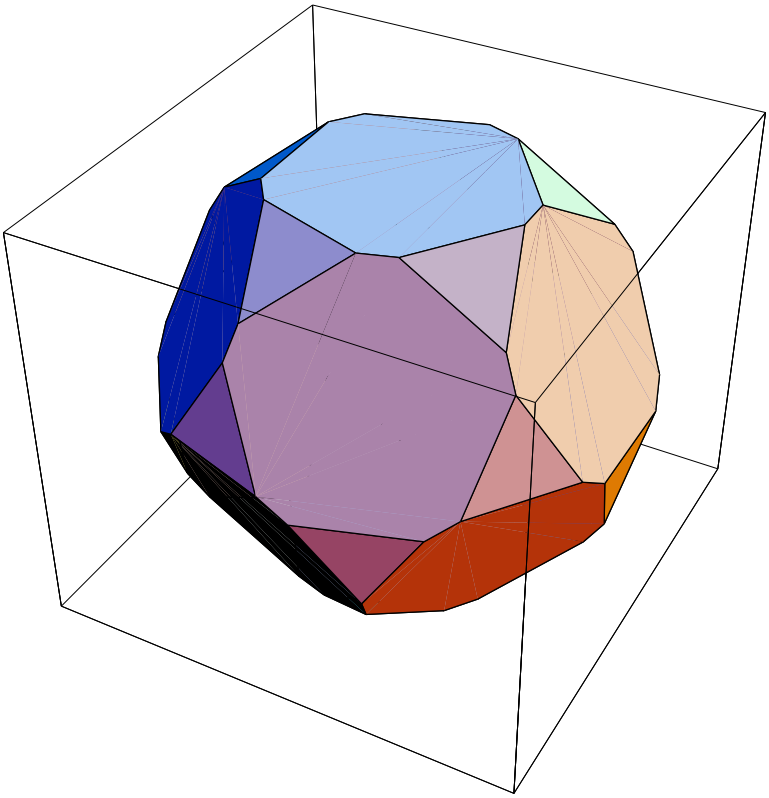


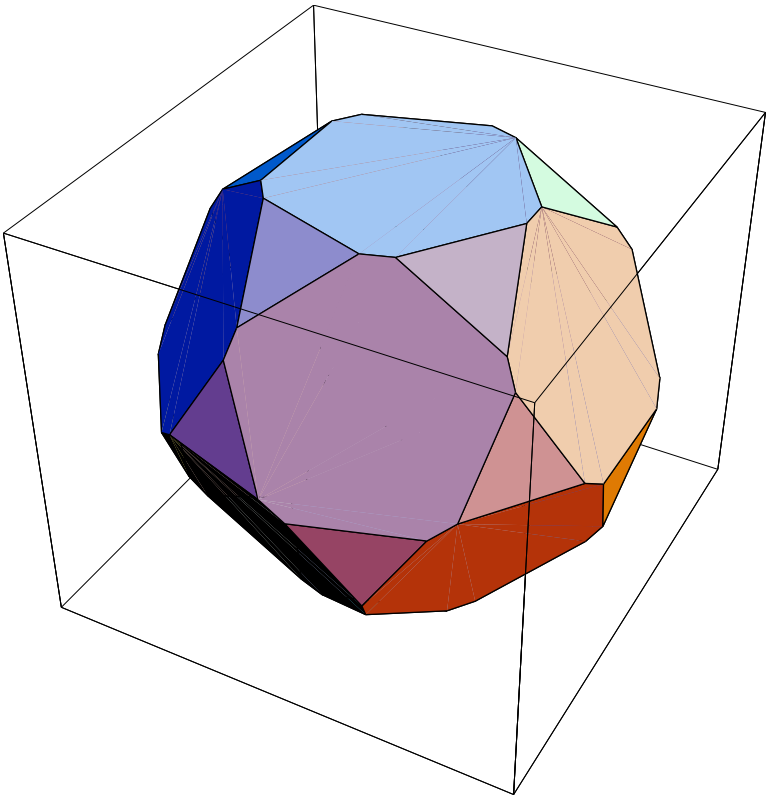


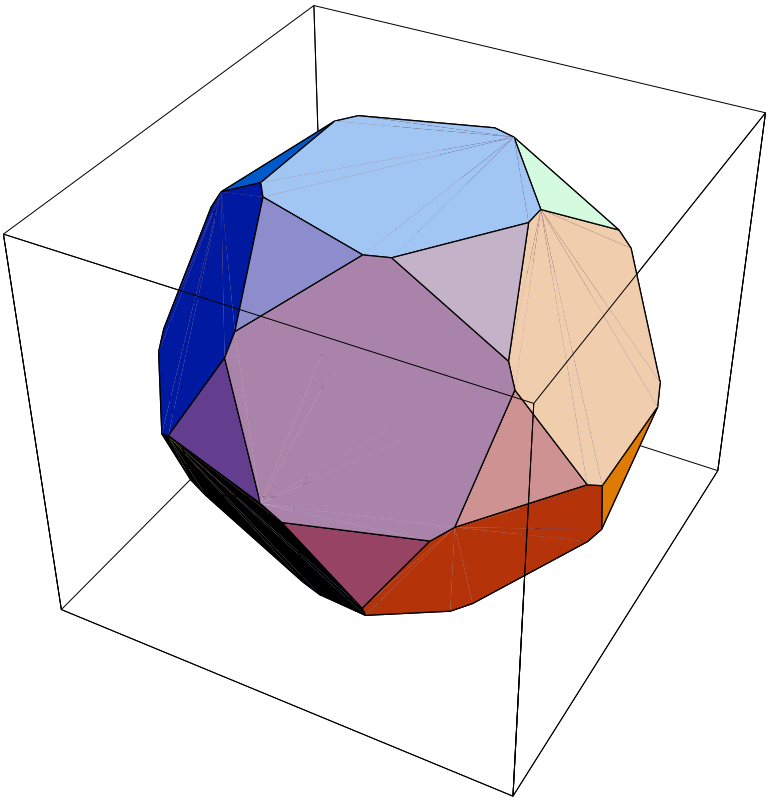


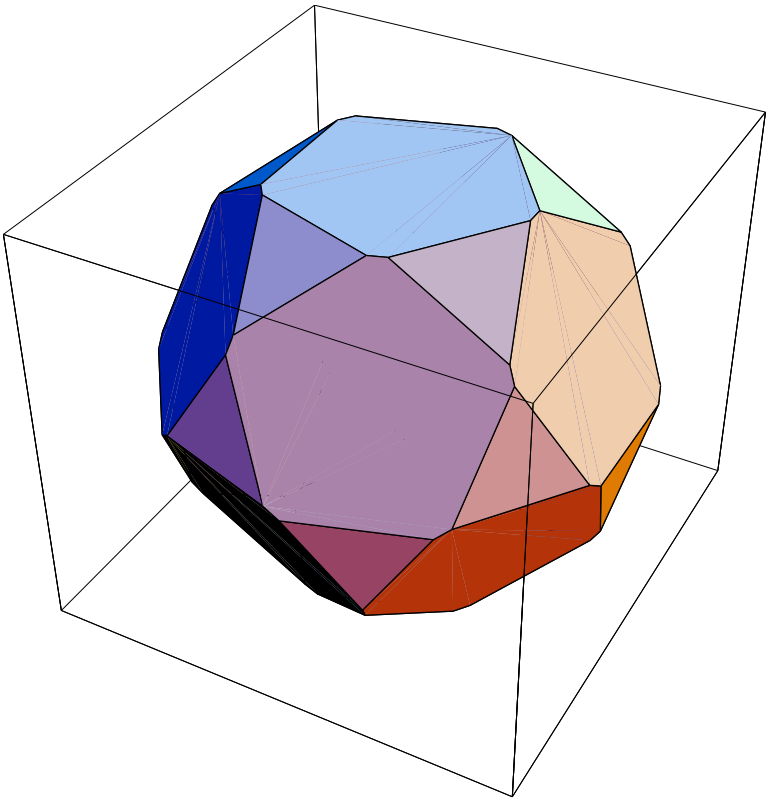


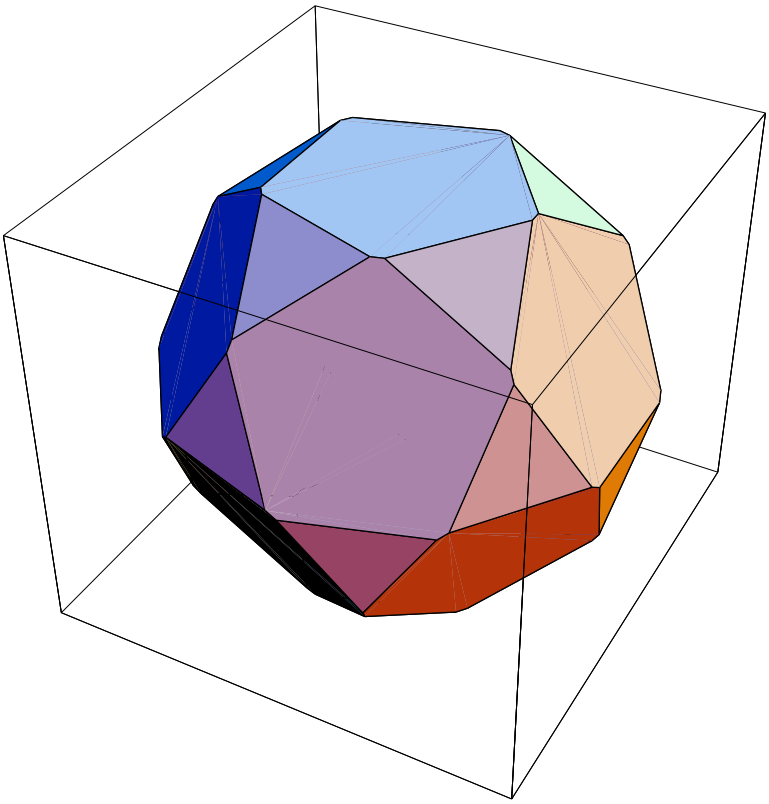


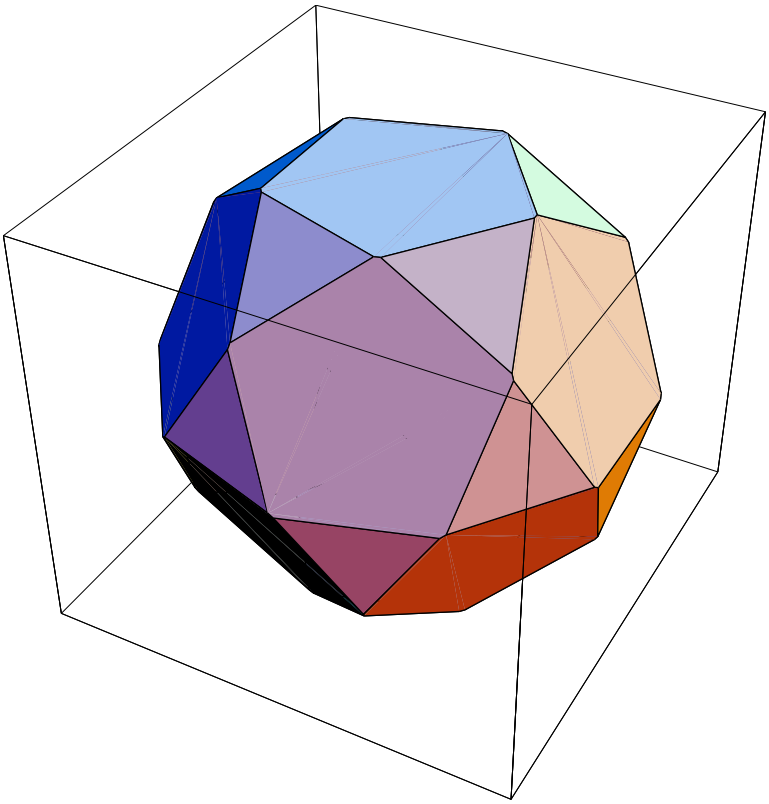


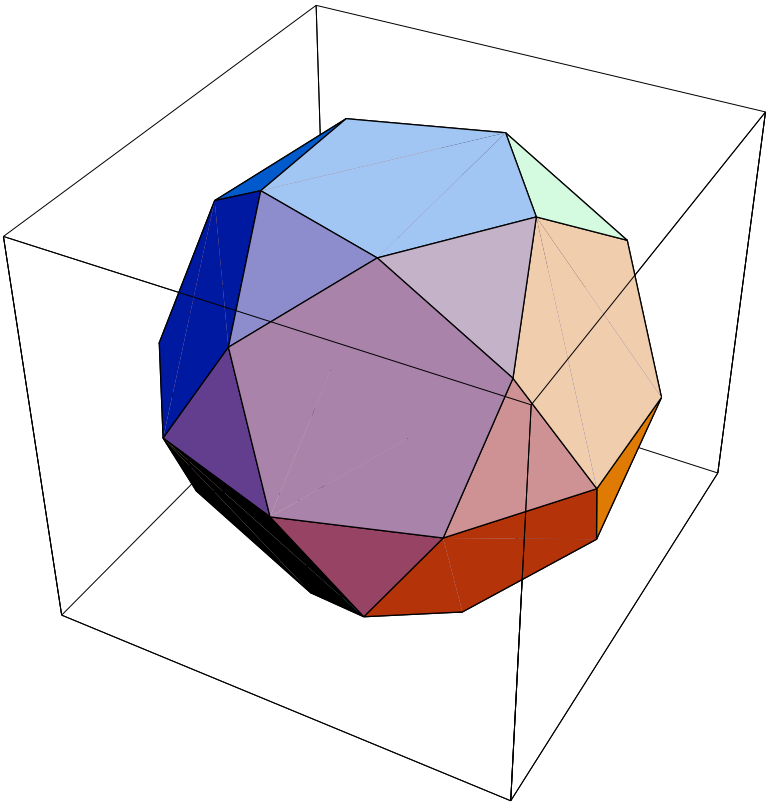




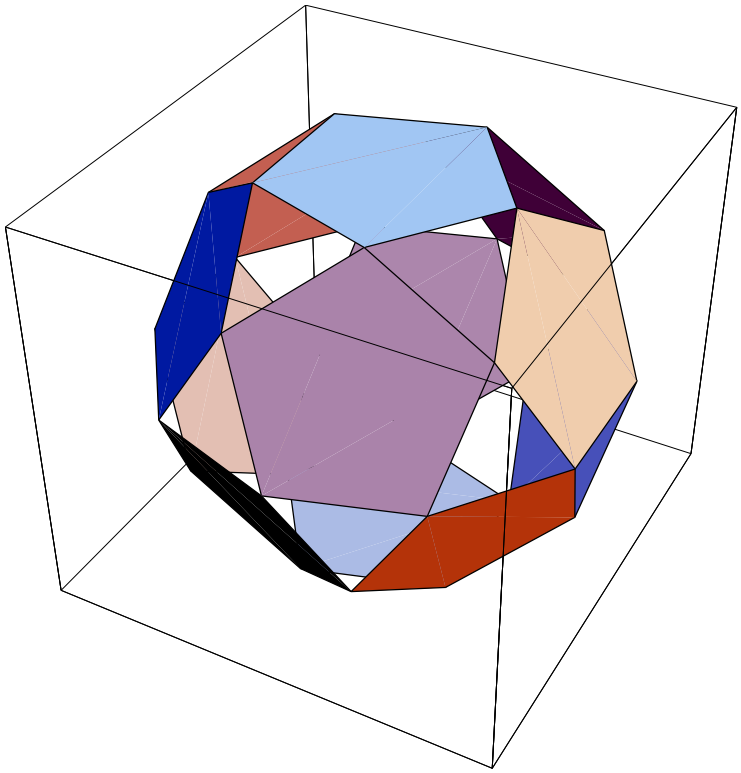


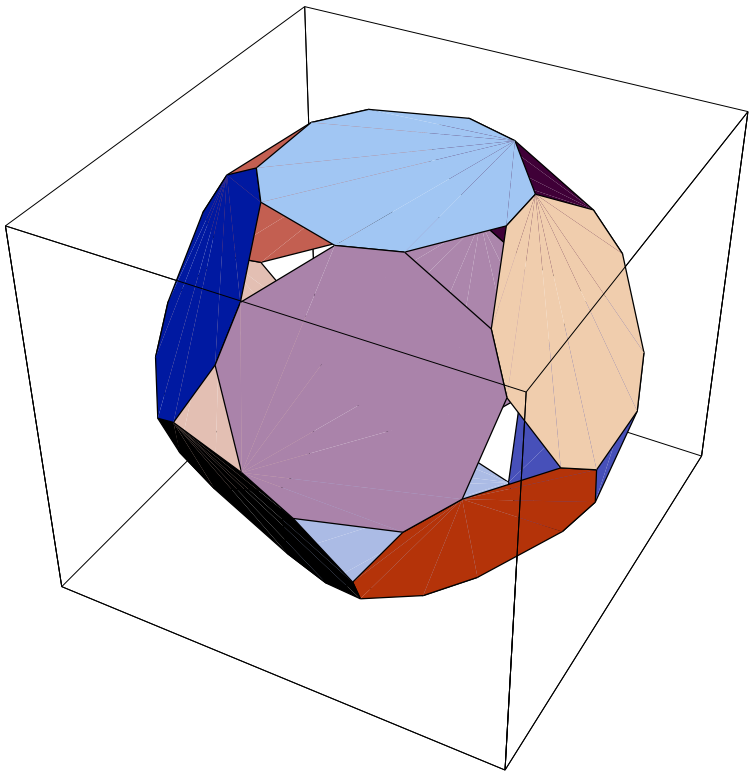


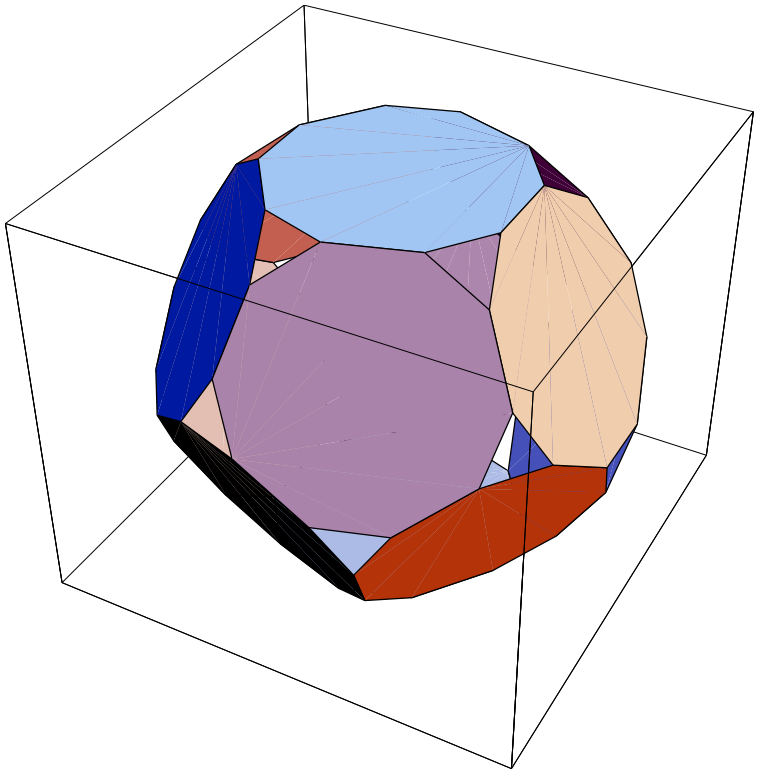


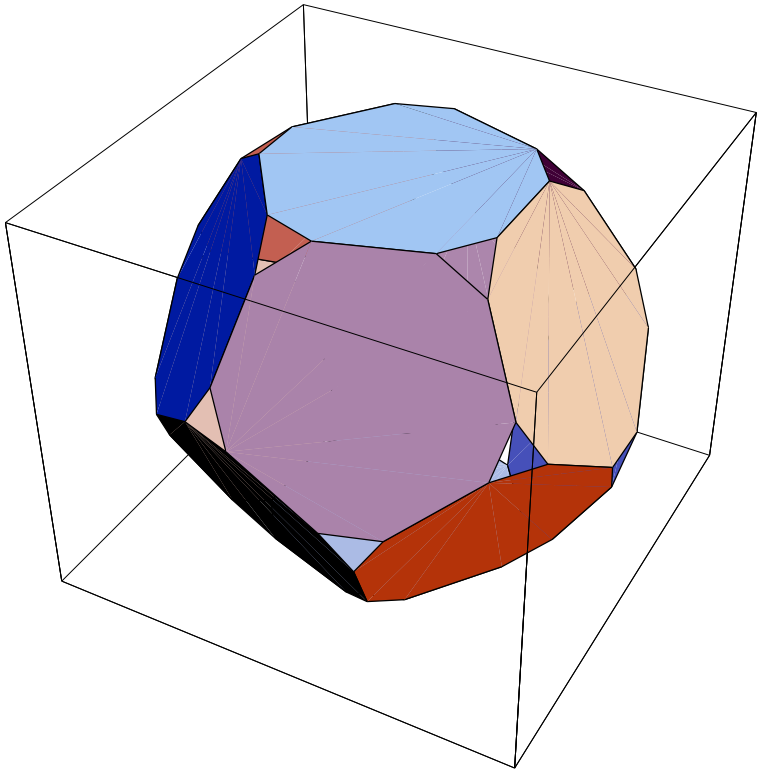


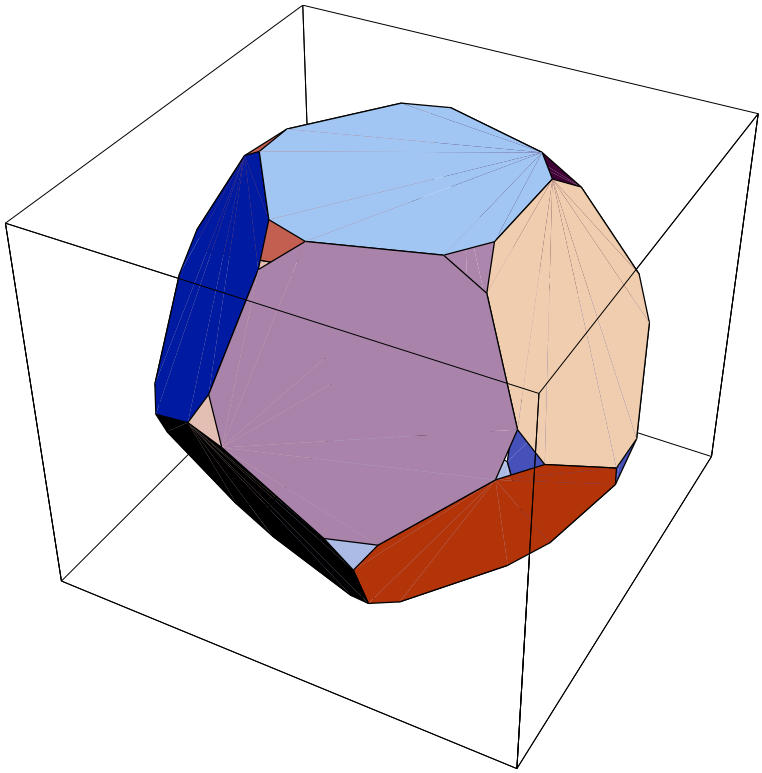
In[32]:=
```
Table[Show[OpenTruncate[Polyhedron[Dodecahedron], 1.0 / k]], {k, 2, 15}];
```

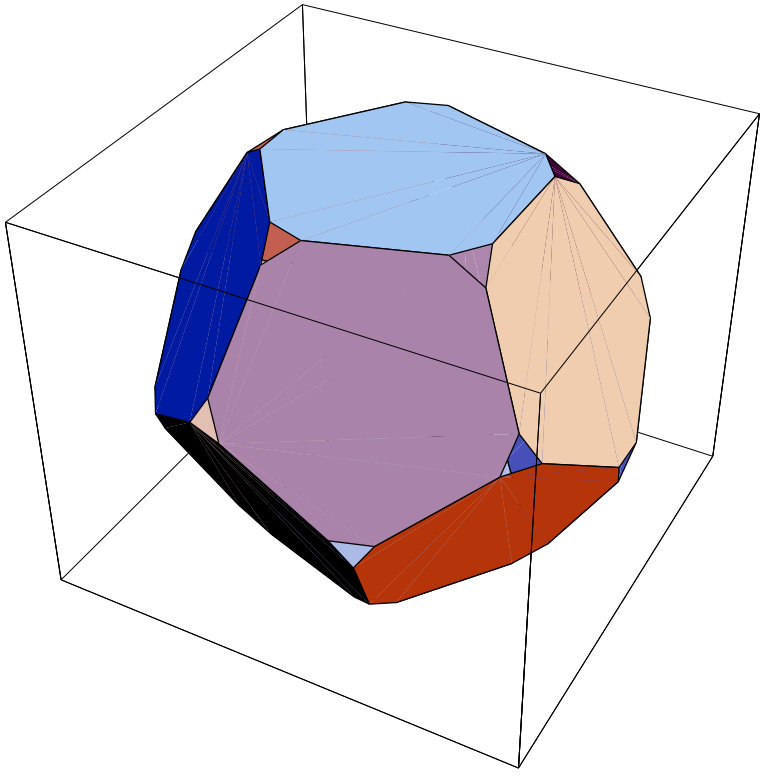


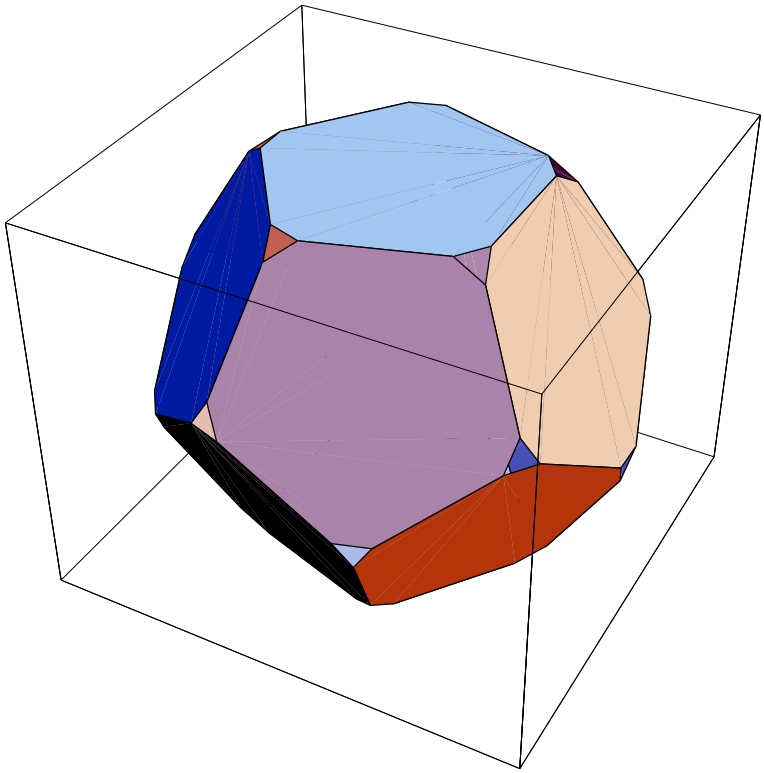


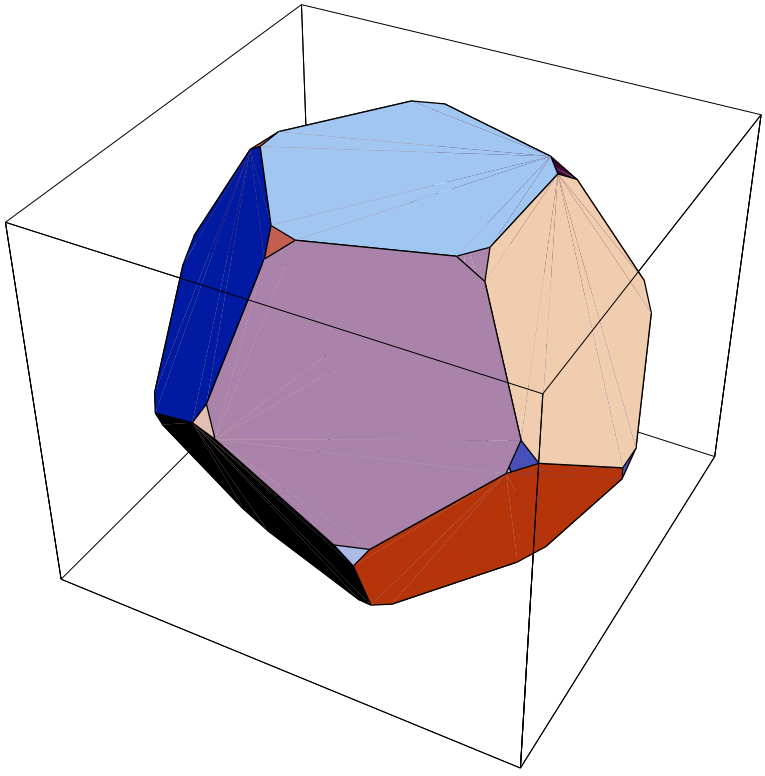


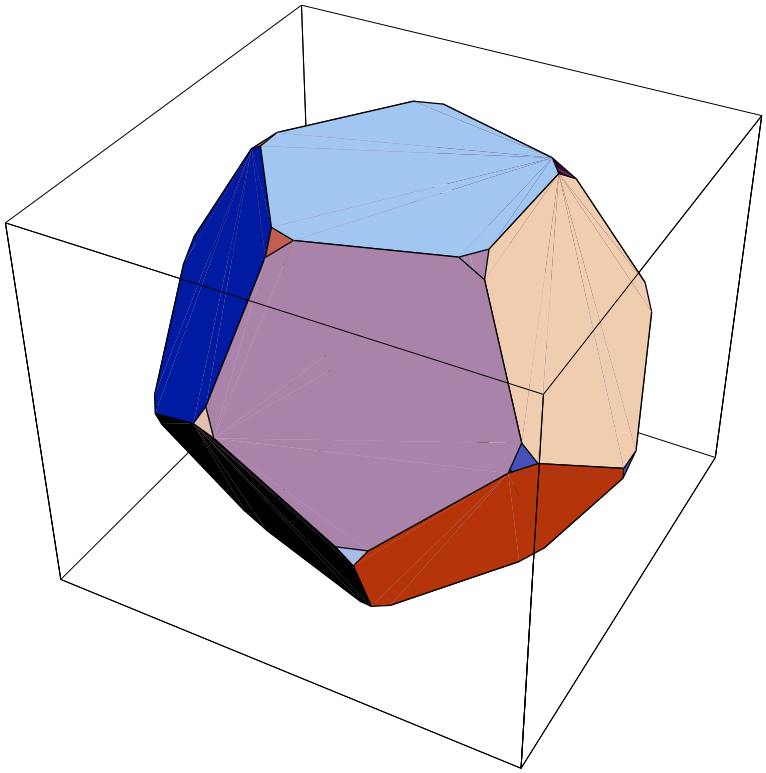


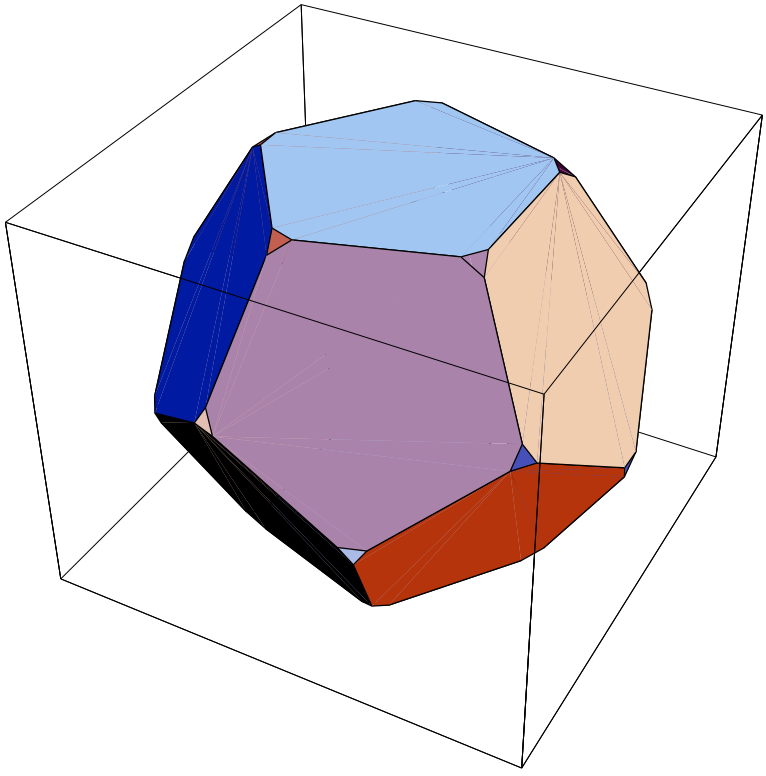


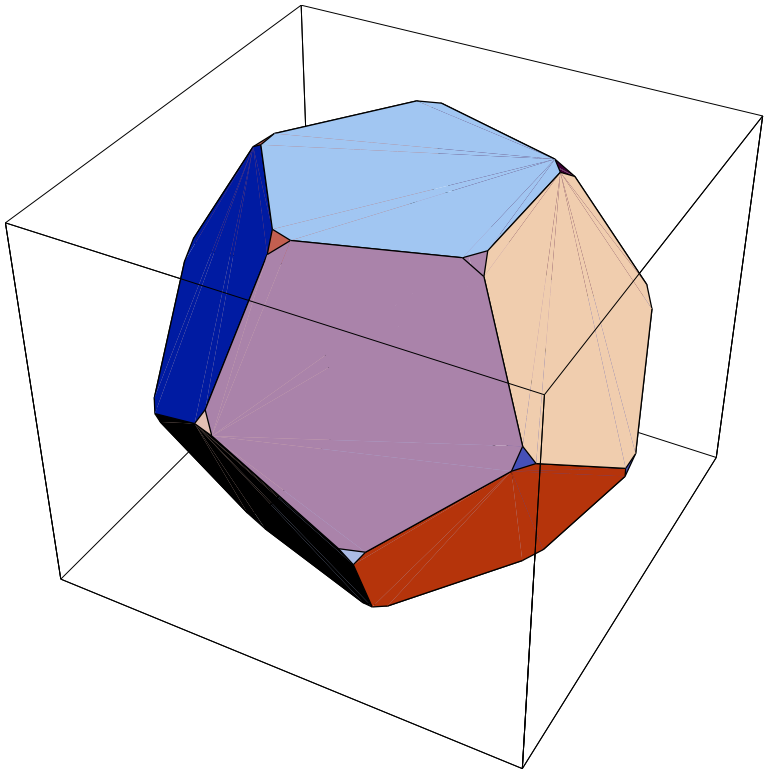


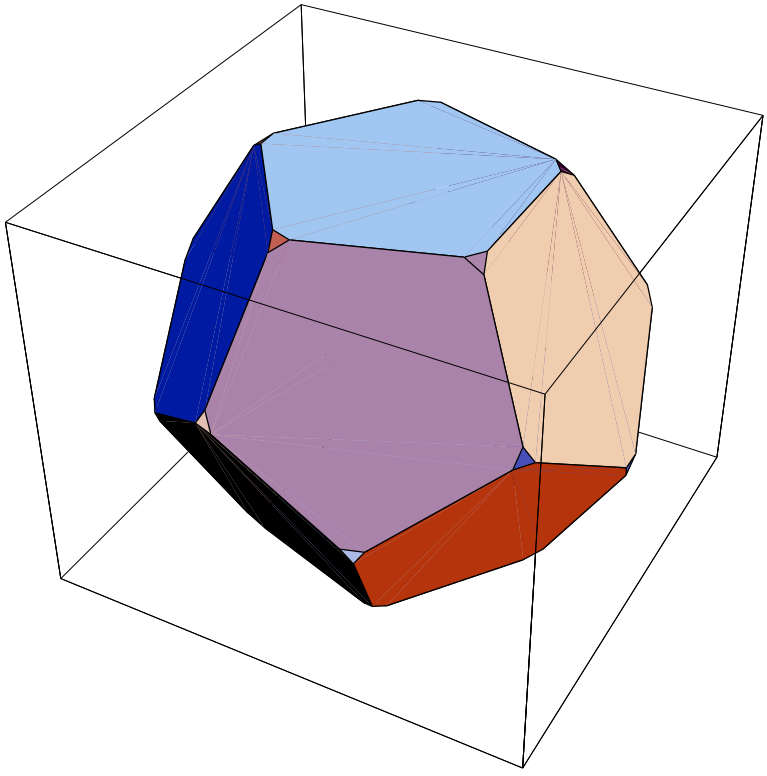


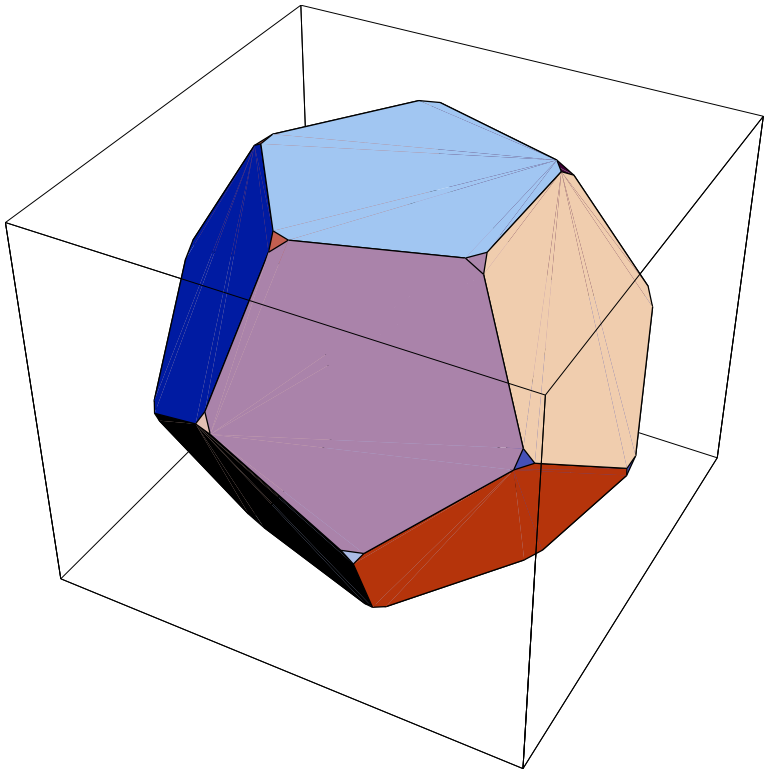


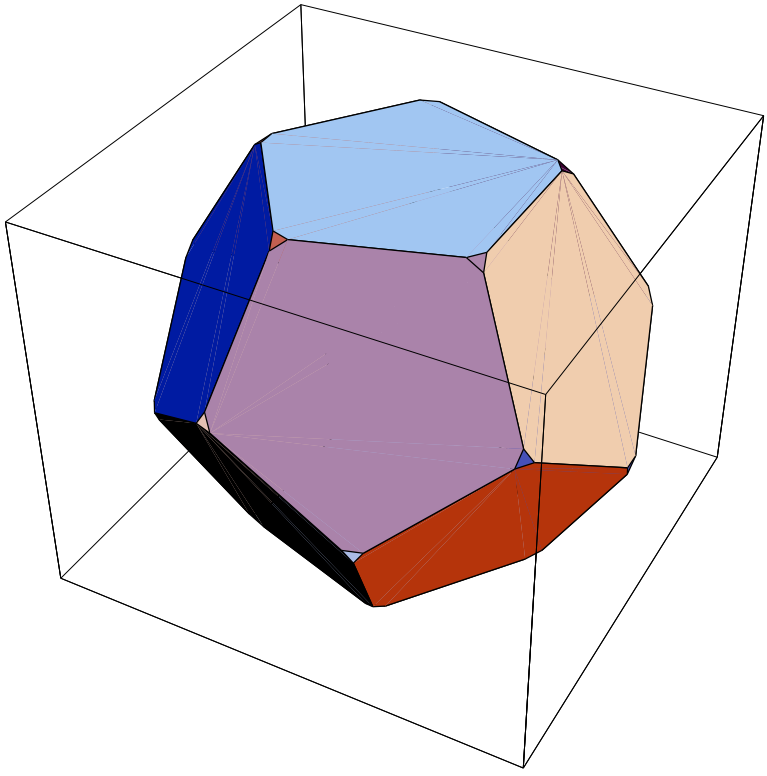




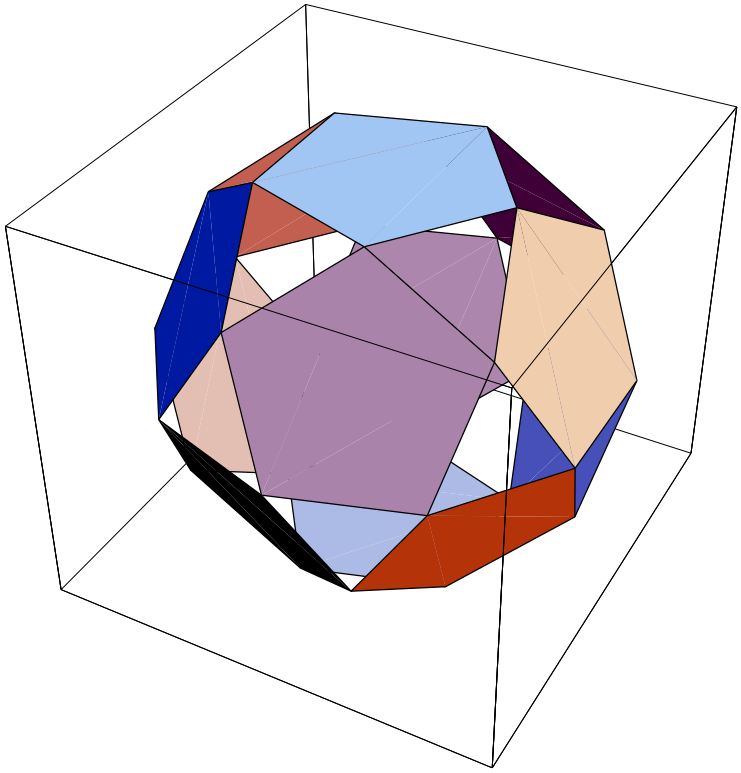




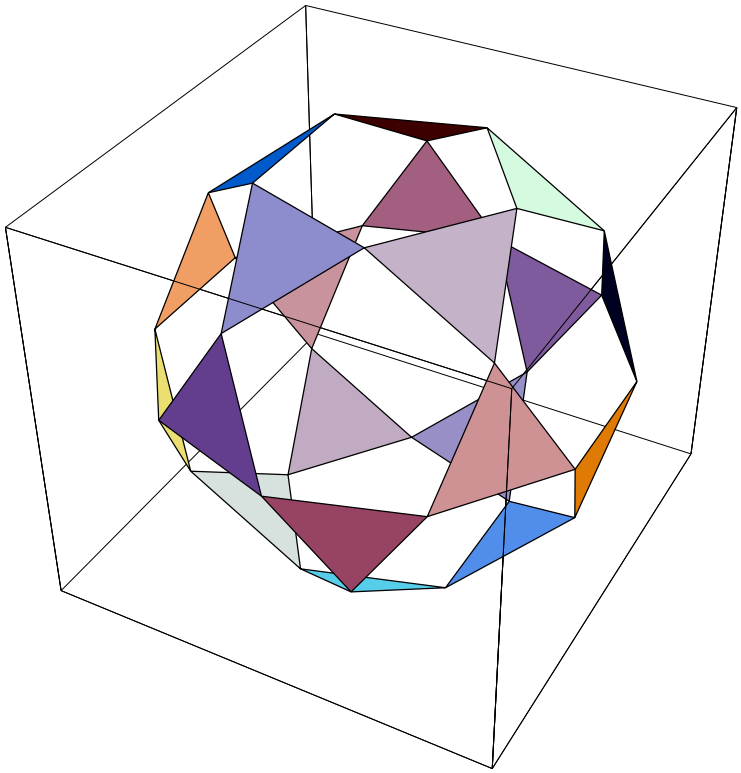




In[33]:=
```
Show[OpenTruncate[Polyhedron[Dodecahedron], 1.0 / 2]];
```

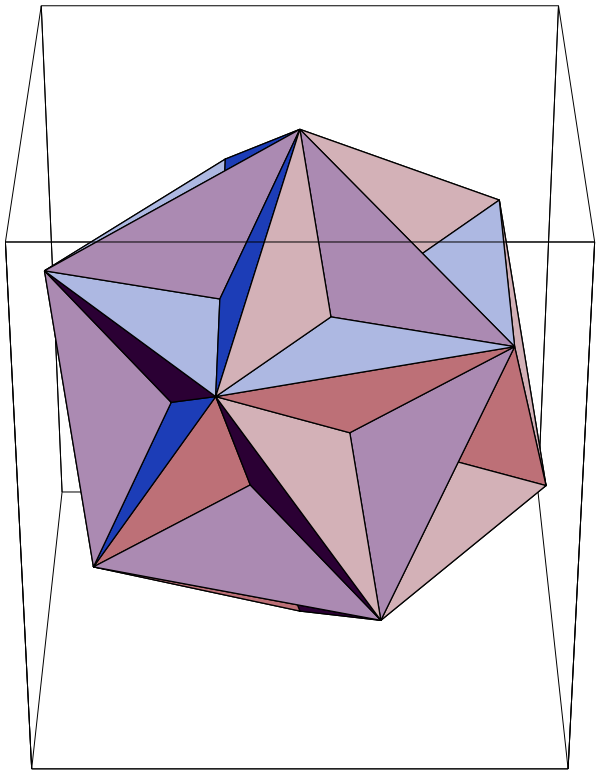


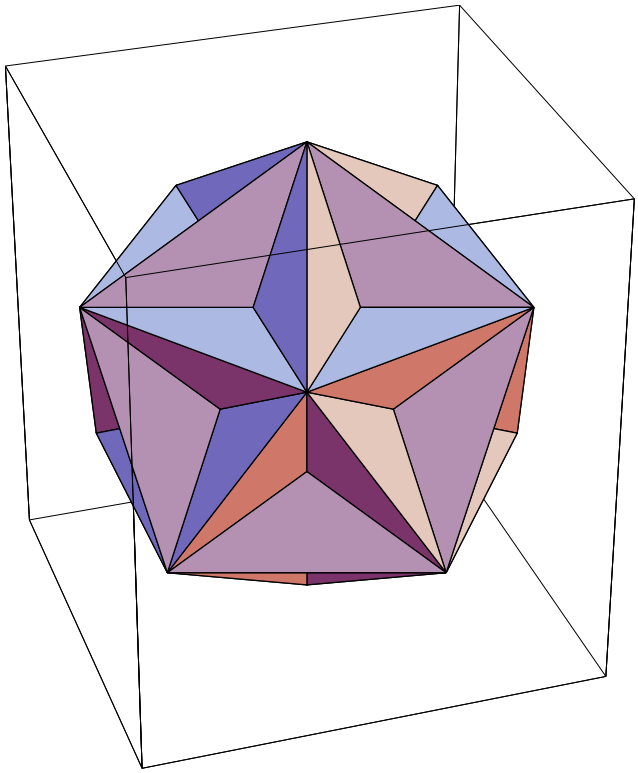
In[34]:=
```
Show[OpenTruncate[Polyhedron[Icosahedron], 1.0 / 2]];
```

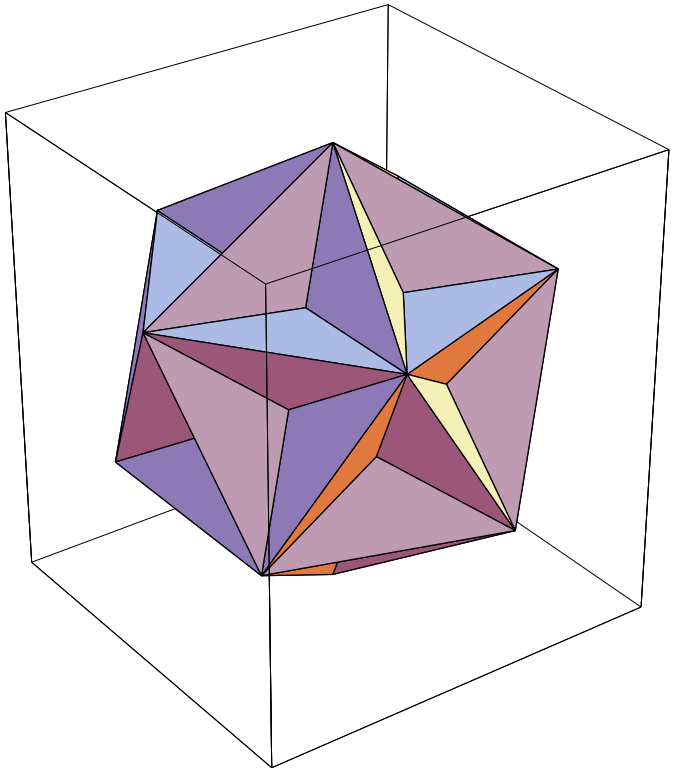


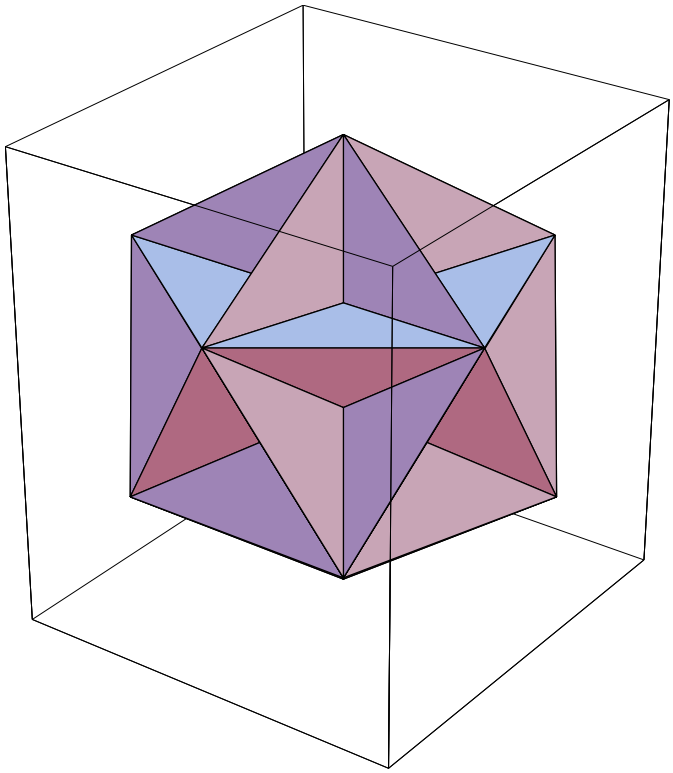
## ■ ViewPoint-Animation: Flug über das Objekt

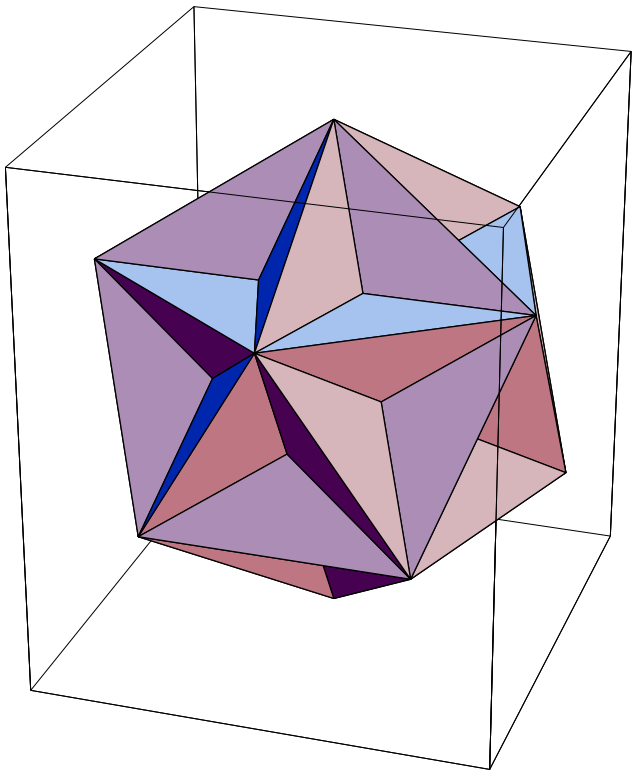
In[35]:=
```
Table[Show[Polyhedron[GreatDodecahedron],
   ViewPoint → {5 Sin[2 Pi / 20 k], 5 Cos[2 Pi / 20 k], 2 + Cos[2 Pi / 20 k]}],
  {k, 0, 19}];
```

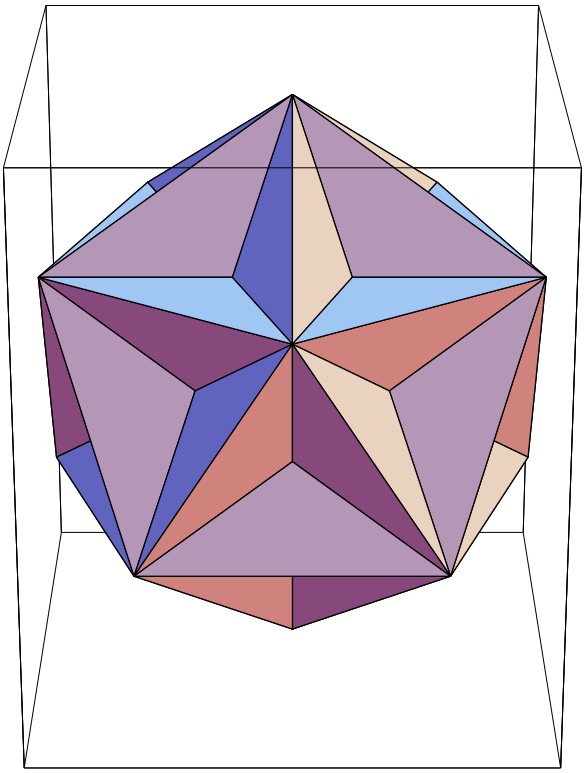


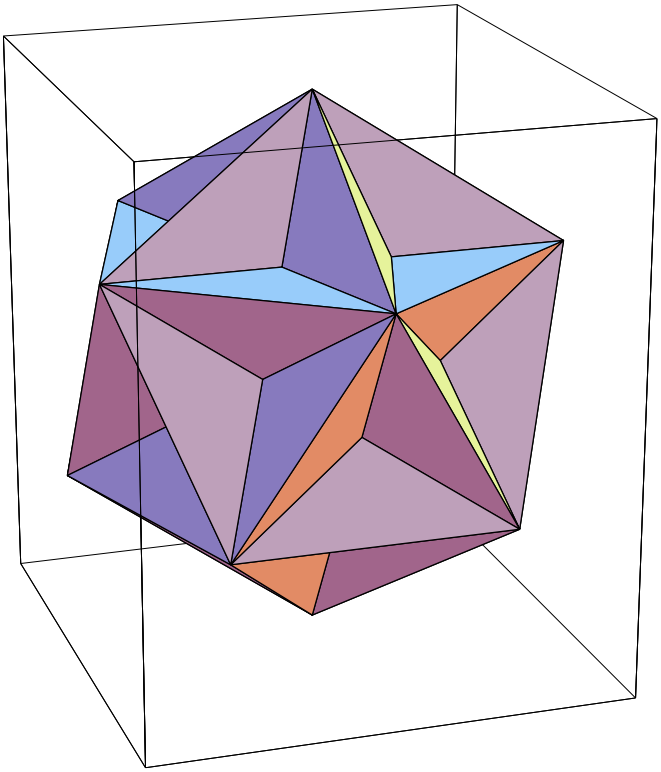


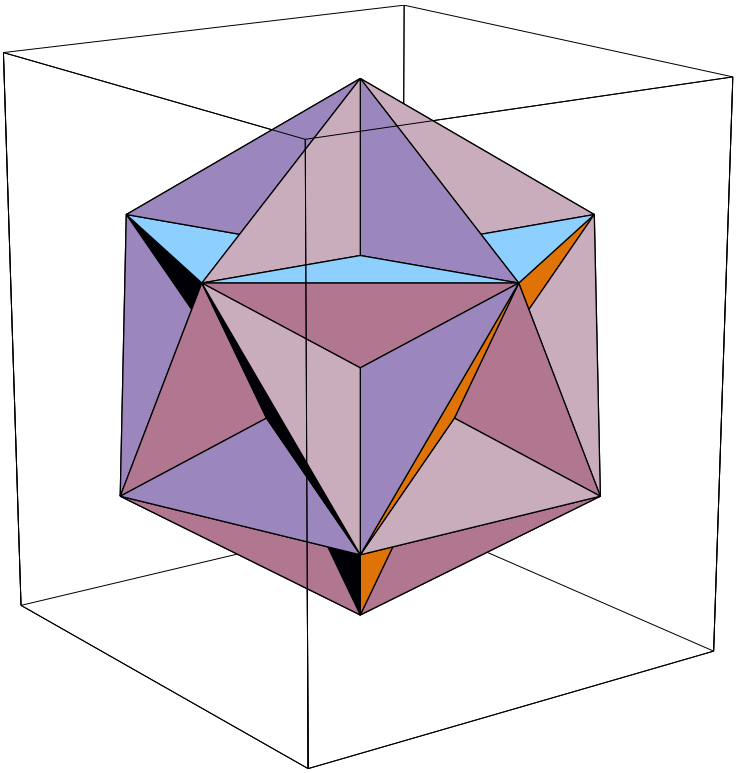


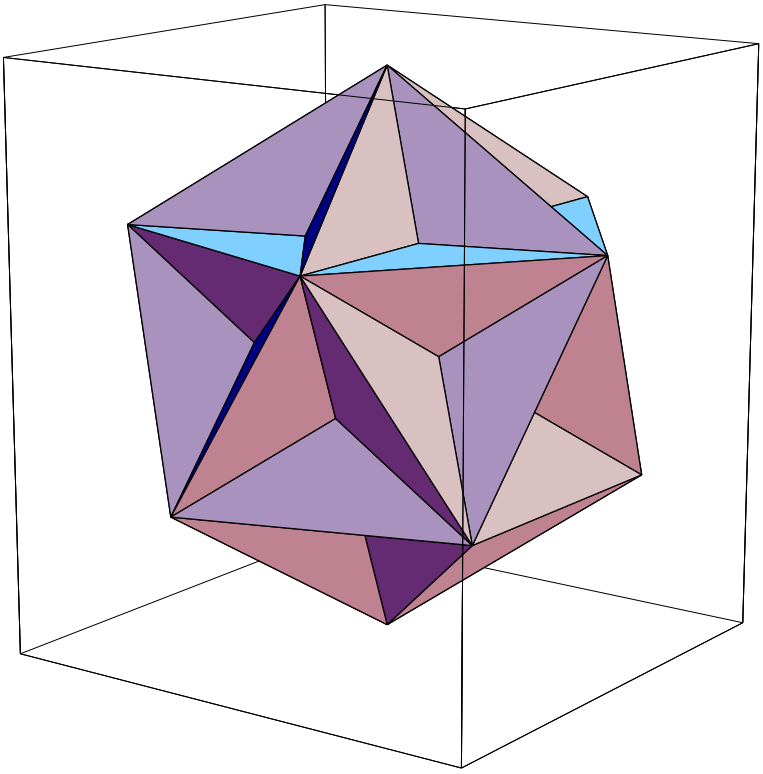


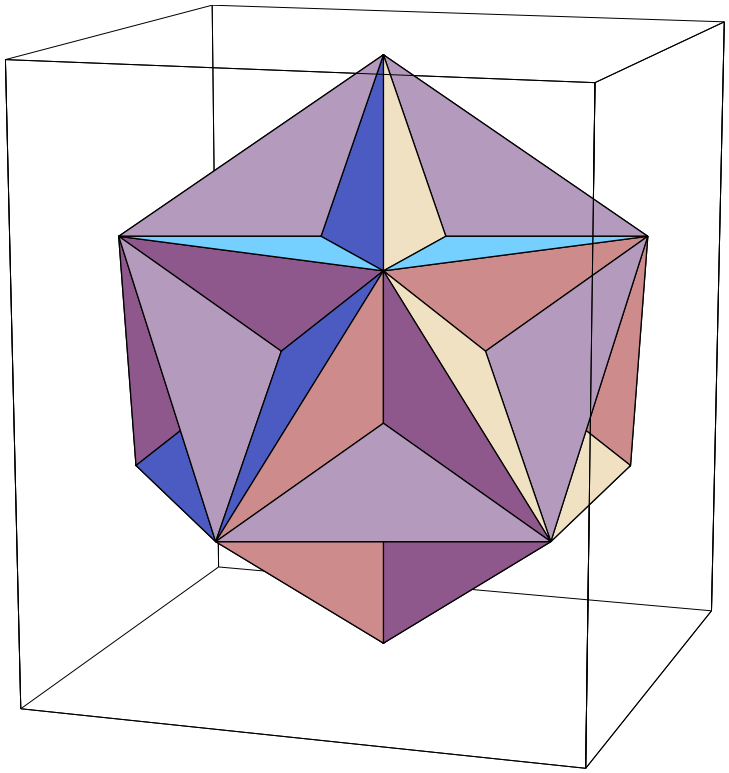


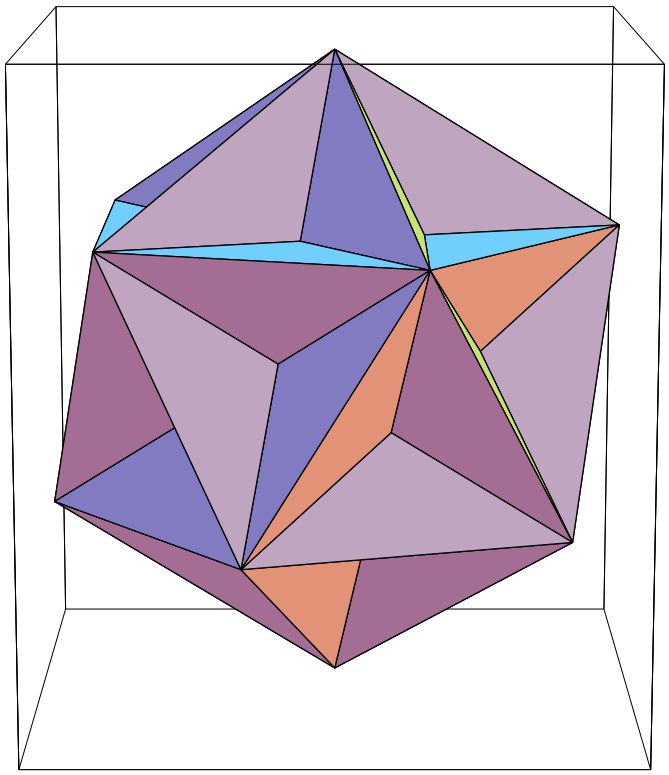


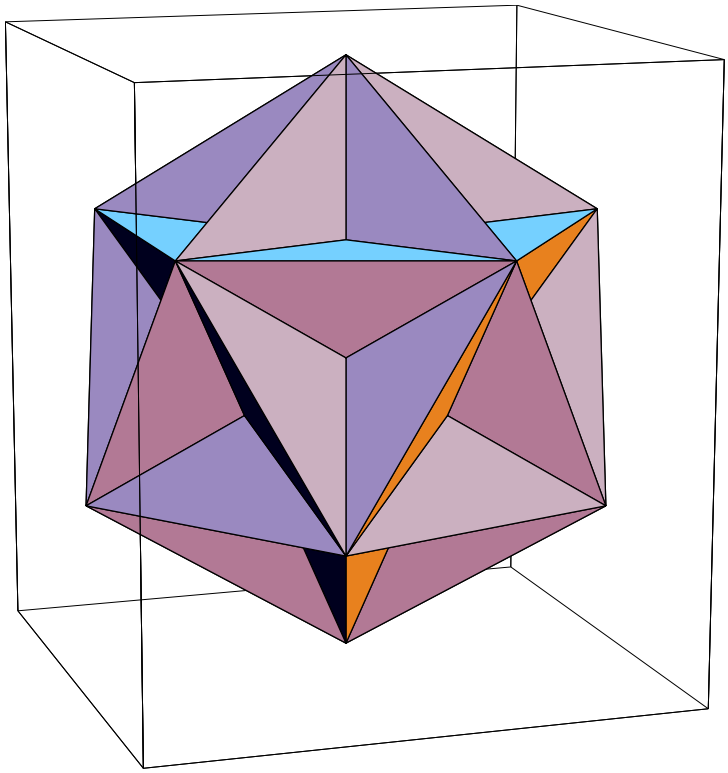


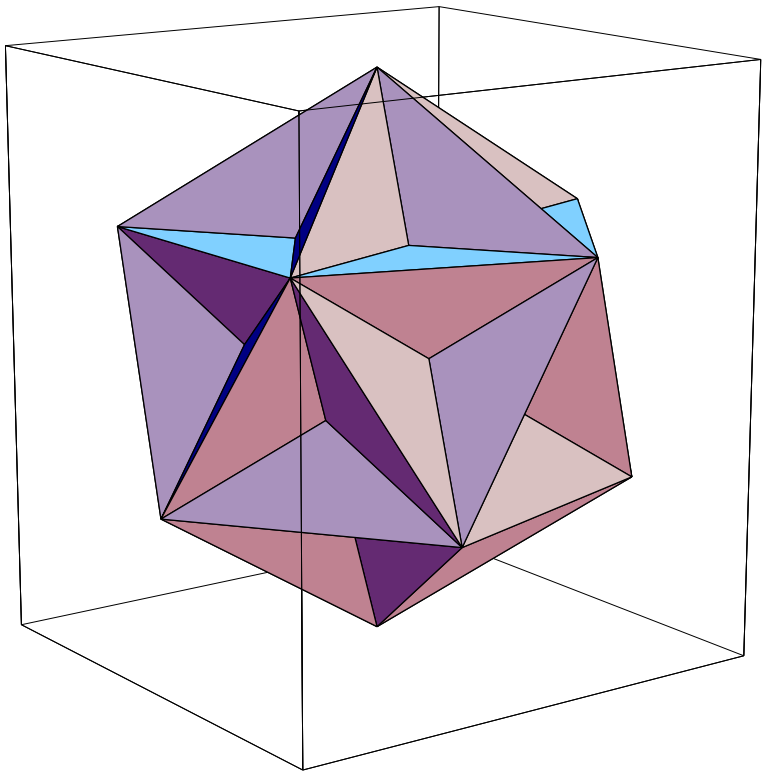


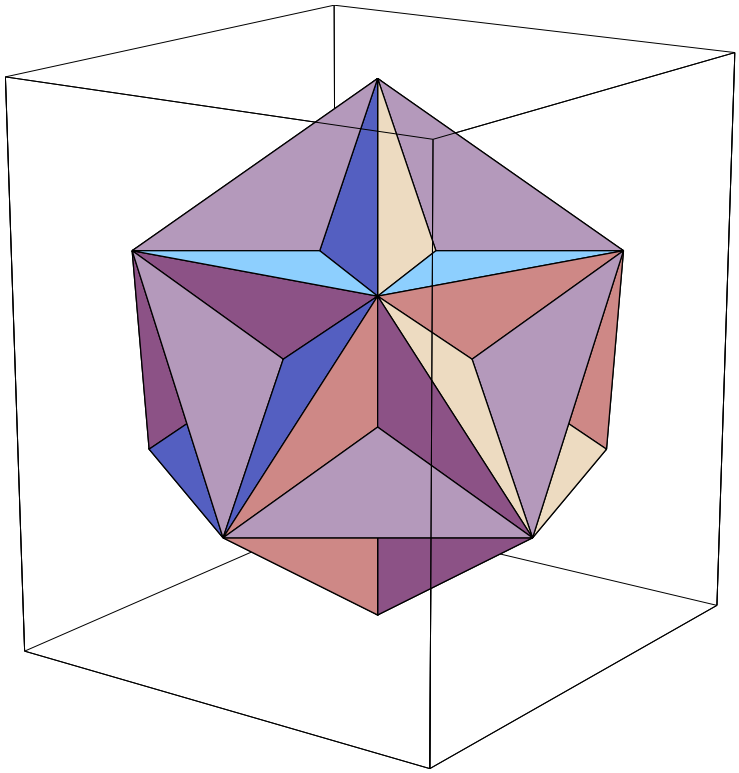


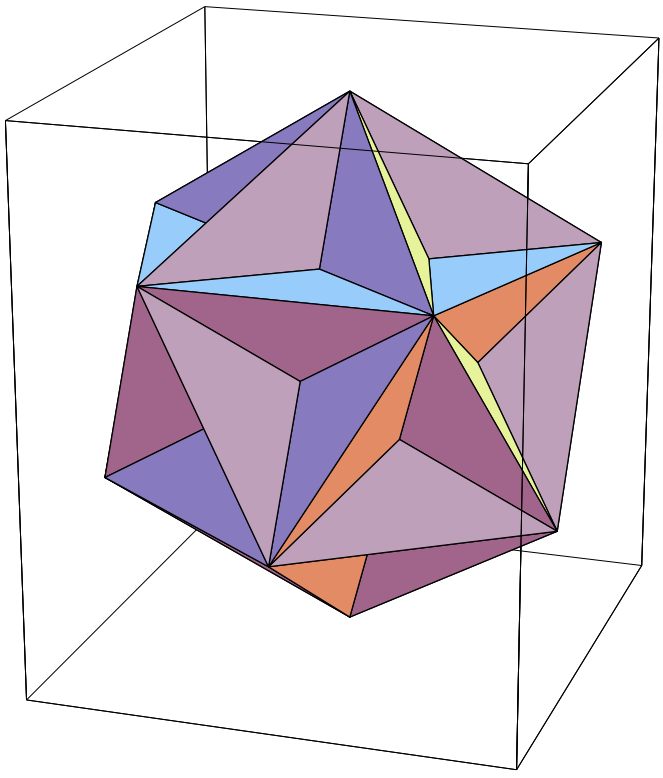


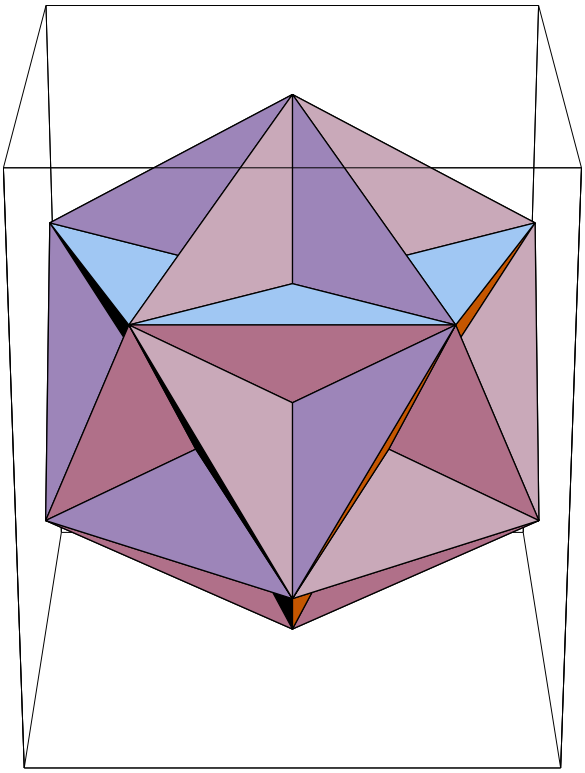


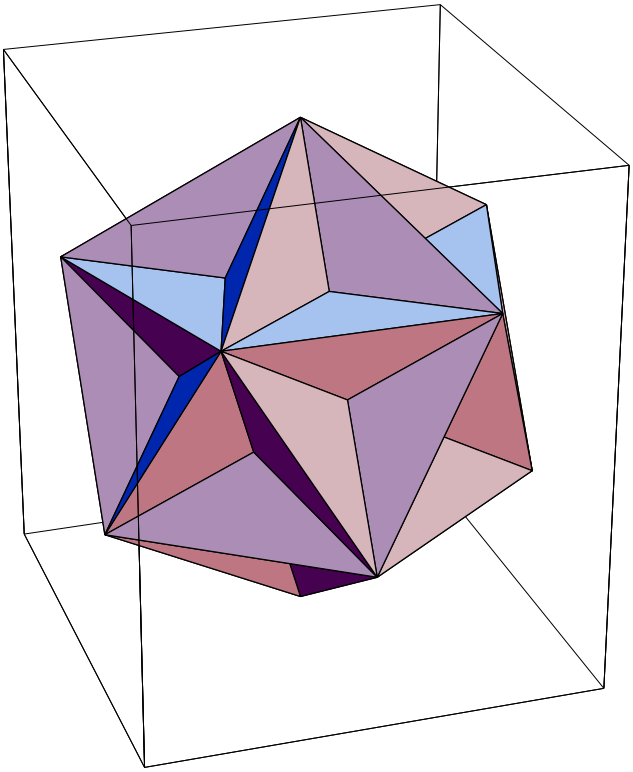


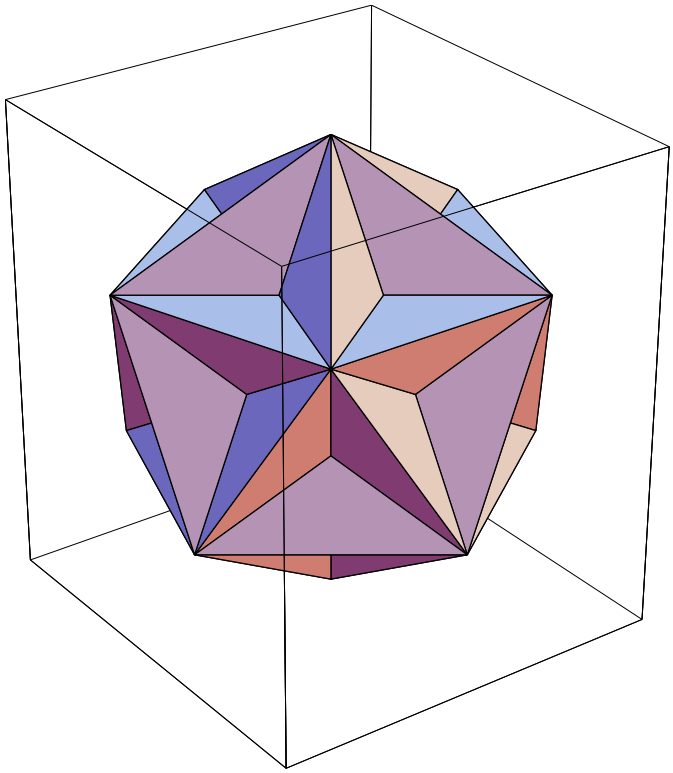


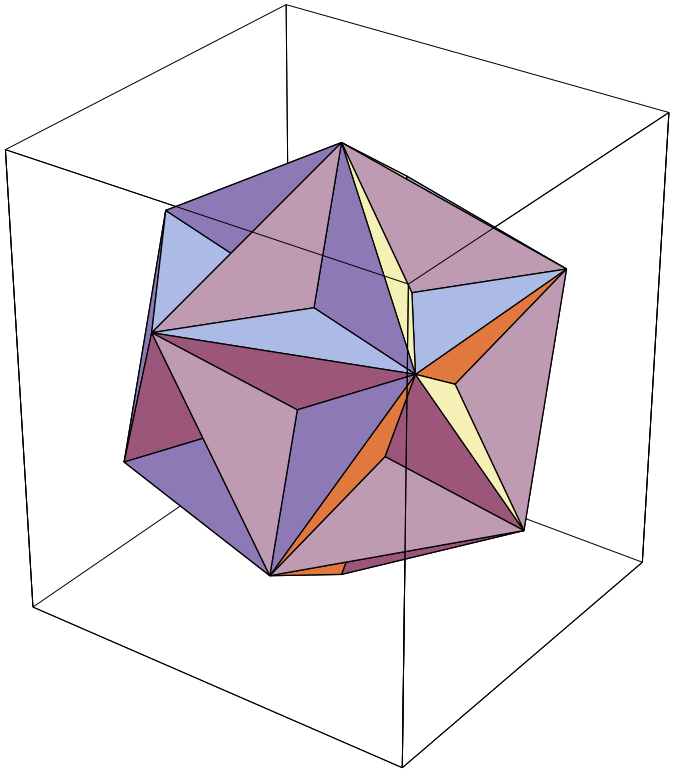


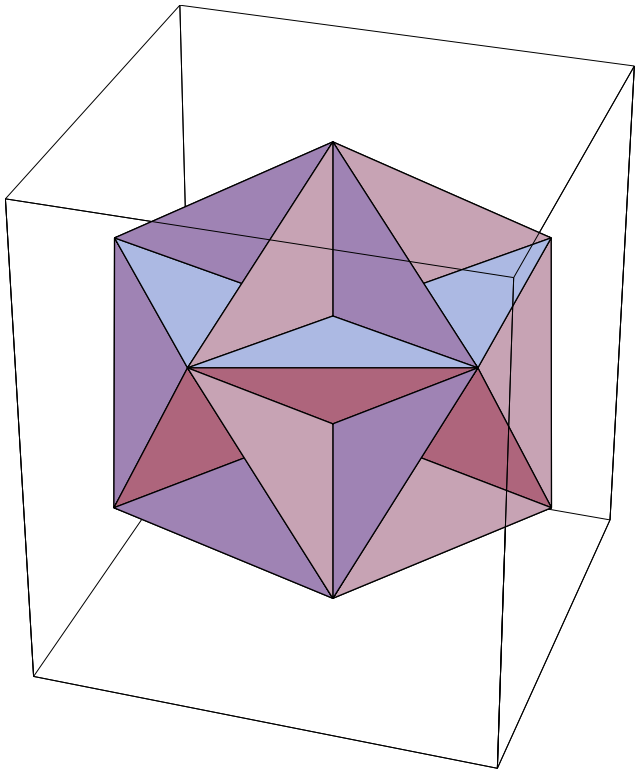












## Package laden zu Polytopen (n-Ecke)

In[36]:=
```
<< Geometry`Polytopes`
```

In[37]:=
```
? Geometry`Polytopes`*
```

### Geometry`Polytopes`

| | | | |
|---|---|---|---|
| Area | Dual | Nonagon | Schlafli |
| CircumscribedRadius | Faces | NumberOfEdges | Square |
| Cube | Heptagon | NumberOfFaces | Tetrahedron |
| Decagon | Hexagon | NumberOfVertices | Triangle |
| Digon | Hexahedron | Octagon | Undecagon |
| Dodecagon | Icosahedron | Octahedron | Vertices |
| Dodecahedron | InscribedRadius | Pentagon | Volume |

In[38]:=
```
$Packages
```

Out[38]=
```
{Geometry`Polytopes`, Utilities`FilterOptions`,
 Graphics`Polyhedra`, Global`, System`}
```

In[39]:=
```
$Path
```

Out[39]=
```
{C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\
   Mathematica\Kernel, C:\Dokumente und Einstellungen\
   Rolf\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\
   Mathematica\Applications, C:\Dokumente und Einstellungen\
   All Users\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
   Mathematica\Autoload, C:\Dokumente und Einstellungen\
   All Users\Anwendungsdaten\Mathematica\Applications,
 ., C:\Dokumente und Einstellungen\Rolf,
 C:\Programme\Wolfram Research\Mathematica\5.2\
   AddOns\StandardPackages, C:\Programme\Wolfram Research\
   Mathematica\5.2\AddOns\StandardPackages\StartUp,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
 C:\Programme\Wolfram Research\
   Mathematica\5.2\SystemFiles\Graphics\Packages,
 C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}
```

In[40]:=
```
$BaseDirectory
```

Out[40]=
```
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica
```

In[41]:=
```
$UserBaseDirectory
```

Out[41]=
```
C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica
```

## Flächen

In[42]:=
```
<< Graphics`Shapes`
```

In[43]:=
```
? Graphics`Shapes`*
```

**Graphics`Shapes`**

| | | | |
|---|---|---|---|
| AffineShape | Helix | RotateShape | TranslateShape |
| Cone | MoebiusStrip | ShrinkPolygons | WireFrame |
| Cylinder | OutlinePolygons | Sphere | |
| DoubleHelix | PerforatePolygons | Torus | |

# B. Eigenbau, Verständnis der Zusammenhänge beim Generieren, mit vielen Entwicklungsschritten, die später weggelassen werden können

Achtung: Warnmeldungen wegen ähnlichen Variablennamen, nicht gefährlich!

## Darstellungsarten:

**Eckenmodelle**

**Kantenmodelle**

**Diagonalenmodelle**

**Aussenflächenmodelle**

**Symmetreieflächenmodelle**

**Sehnenflächenmodelle**

**Sternformenmodelle**

**Durchdringungsmodelle**

**Schnittmodelle**

**Hüllenmodelle von Sternformen**

**Kombinationen solcher Möglichkeiten**

**Projektionen, Animationen, .....**

**Abwicklungen**

**Modelle nach eigene Ideen**

## Einfache Möglichkeit: Würfel aus dem Prisma heraus

## Berechnungen zum Würfel:

sW = Kantenlänge

In[44]:=
```
volW = sW^3;
diagFW = Sqrt[2] sW;
diagVW = Sqrt[3] sW;
rUW = diagVW / 2;
rIW = sW / 2;
 rKW = diagVW / 2;
obW = 6 sW^2;
```

## Graphik

In[50]:=
```
pB[k_] := {(-1)^(Floor[(k + 1) / 2]), (-1)^(Floor[k / 2])}
```

In[51]:=
```
taG = Table[pB[k], {k, 0, 3}]
```

Out[51]=
```
{{1, 1}, {-1, 1}, {-1, -1}, {1, -1}}
```

In[52]:=
```
lift[k_, vec_] := Append[vec, k]
```

In[53]:=
```
taW = Flatten[Table[Append[taG[[n]], k], {n, 1, Length[taG]},
   {k, -1, 1, 2}], 1]
```

Out[53]=
```
{{1, 1, -1}, {1, 1, 1}, {-1, 1, -1}, {-1, 1, 1},
 {-1, -1, -1}, {-1, -1, 1}, {1, -1, -1}, {1, -1, 1}}
```

In[54]:=
```
taW = Join[Table[Append[taG[[n]], -1], {n, 1, Length[taG]}],
  Table[Append[taG[[n]], 1], {n, 1, Length[taG]}]]
```

Out[54]=
```
{{1, 1, -1}, {-1, 1, -1}, {-1, -1, -1},
 {1, -1, -1}, {1, 1, 1}, {-1, 1, 1}, {-1, -1, 1}, {1, -1, 1}}
```

In[55]:=
```
wue = Wuerfel; tet = Tetraeder; tetN = TetraederNeg; okt = Oktaeder;
dod = Dodekaeder;
iko = Ikosaeder;
```

In[56]:=
```
p[k_, wue] := taW[[k]];
pointW = Prepend[Table[Point[p[k, wue]], {k, 1, Length[taW]}],
  PointSize[0.03]]
```

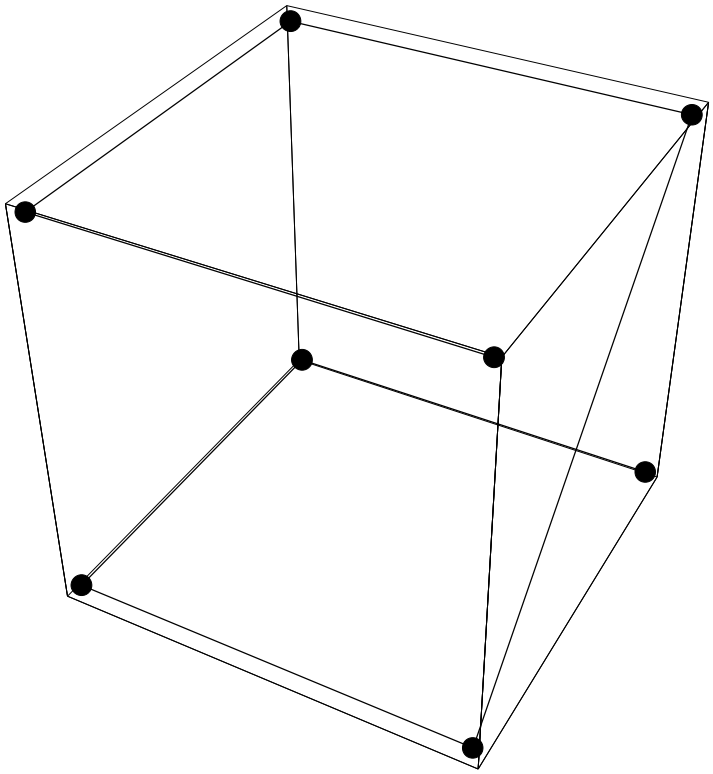
Out[57]=
```
{PointSize[0.03], Point[{1, 1, -1}], Point[{-1, 1, -1}],
 Point[{-1, -1, -1}], Point[{1, -1, -1}], Point[{1, 1, 1}],
 Point[{-1, 1, 1}], Point[{-1, -1, 1}], Point[{1, -1, 1}]}
```

In[58]:=
```
pz[k_, wue, z_] := taW[[k]] z;
pointWz[z_] :=
 Prepend[Table[Point[pz[k, wue, z]], {k, 1, Length[taW]}],
  PointSize[0.03]]
```

In[60]:=
```
lW = Table[p[k, wue], {k, 1, Length[taW]}]
```

Out[60]=
```
{{1, 1, -1}, {-1, 1, -1}, {-1, -1, -1},
 {1, -1, -1}, {1, 1, 1}, {-1, 1, 1}, {-1, -1, 1}, {1, -1, 1}}
```

In[61]:=
```
Show[Graphics3D[{pointW, Line[lW]}]];
```

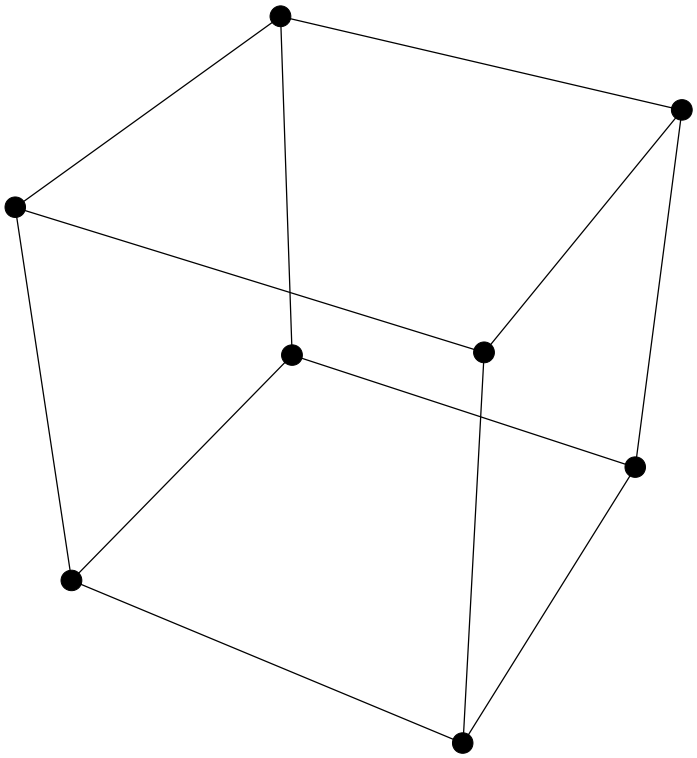


In[62]:=
```
Options[Graphics3D]
```

Out[62]=
```
{AmbientLight → GrayLevel[0], AspectRatio → Automatic,
 Axes → False, AxesEdge → Automatic, AxesLabel → None,
 AxesStyle → Automatic, Background → Automatic, Boxed → True,
 BoxRatios → Automatic, BoxStyle → Automatic, ColorOutput → Automatic,
 DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
 DisplayFunction :→ $DisplayFunction, Epilog → {}, FaceGrids → None,
 FormatType :→ $FormatType, ImageSize → Automatic, Lighting → True,
 LightSources → {{{1., 0., 1.}, RGBColor[1, 0, 0]},
   {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}},
 Plot3Matrix → Automatic, PlotLabel → None, PlotRange → Automatic,
 PlotRegion → Automatic, PolygonIntersections → True, Prolog → {},
 RenderAll → True, Shading → True, SphericalRegion → False,
 TextStyle :→ $TextStyle, Ticks → Automatic, ViewCenter → Automatic,
 ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

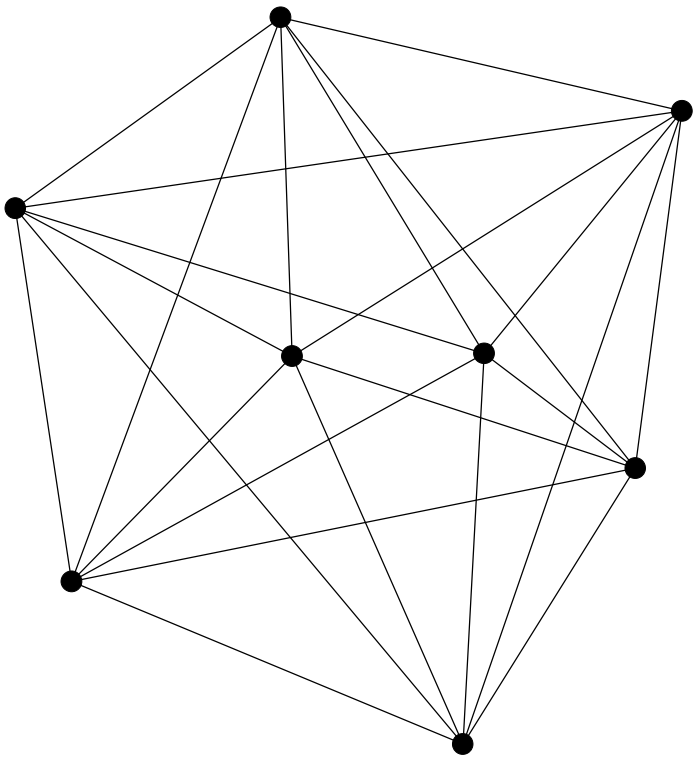
In[63]:=
```
lin2[n_, m_, u_] := Line[{p[n, u], p[m, u]}];
lin3[n_, m_, k_, u_] := Line[{p[n, u], p[m, u], p[k, u]}];
lin4[n_, m_, k_, s_, u_] := Line[{p[n, u], p[m, u], p[k, u], p[s, u]}];
lin5[n_, m_, k_, s_, t_, u_] :=
 Line[{p[n, u], p[m, u], p[k, u], p[s, u], p[t, u]}];
lin6[n_, m_, k_, s_, t_, v_, u_] :=
 Line[{p[n, u], p[m, u], p[k, u], p[s, u], p[t, u], p[v, u]}];
linz2[n_, m_, u_, z_] := Line[{p[n, u], p[m, u]} z];
linz3[n_, m_, k_, u_, z_] := Line[{p[n, u], p[m, u], p[k, u]} z];
linz4[n_, m_, k_, s_, u_, z_] :=
 Line[{p[n, u], p[m, u], p[k, u], p[s, u]} z];
linz5[n_, m_, k_, s_, t_, u_, z_] :=
 Line[{p[n, u], p[m, u], p[k, u], p[s, u], p[t, u]} z];
```

In[65]:=
```
wuerfel1 =
  Show[Graphics3D[{pointW, lin5[1, 2, 3, 4, 1, wue],
     lin5[5, 6, 7, 8, 5, wue], lin2[1, 5, wue], lin2[2, 6, wue],
     lin2[3, 7, wue], lin2[4, 8, wue]}, Axes → False, AxesEdge → None,
    Boxed → False]];
```



In[66]:=
```
nimm1 = wuerfel1;
```

In[67]:=
```
wuerfelF1 =
  Show[Graphics3D[{pointW, lin5[1, 2, 3, 4, 1, wue],
     lin5[5, 6, 7, 8, 5, wue], lin2[1, 5, wue], lin2[2, 6, wue],
     lin2[3, 7, wue], lin2[4, 8, wue], lin2[1, 3, wue], lin2[2, 4, wue],
     lin2[1, 6, wue], lin2[2, 5, wue],
     lin2[2, 7, wue], lin2[3, 6, wue],
     lin2[3, 8, wue], lin2[4, 7, wue],
     lin2[4, 5, wue], lin2[1, 8, wue],
     lin2[5, 7, wue], lin2[6, 8, wue]},
    Axes → False, AxesEdge → None, Boxed → False]];
```



## Tetraeder aus dem Würfel heraus

### Berechnungen zum Tetraeder:

In[68]:=
```
{Sqrt[1^2 - (1 / 2)^2], Sqrt[1^2 - (2 / 3 Sqrt[3] / 2)^2]}
```

Out[68]=
$$\left\{\frac{\sqrt{3}}{2}, \sqrt{\frac{2}{3}}\right\}$$

sT = Kantenlänge

In[69]:=
```
HoeheFT = sT Sqrt[3] / 2; HoeheVT = sT Sqrt[2 / 3];
sT = diagFW; diagFT = HoeheFT; rUT = 2 / 3 HoeheVT; rIT = 1 / 4 * HoeheVT;
rK = Sqrt[rIT^2 + (1 / 3 HoeheFT^2)]; volT = sT * HoeheFT / 2 * HoeheVT / 3;
obT = 4 * sT * HoeheFT / 2;
volT1 = sW^3 - 4 * sW * sW / 2 * sW / 3;
```

## Graphik zum Tetraeder:

In[72]:=
```
p[k_ /; k < 3, tet] := p[2 k, wue]; p[k_ /; k >= 3, tet] := p[2 k - 1, wue];
p[k_ /; k < 3, tetN] := p[2 k - 1, wue]; p[k_ /; k >= 3, tetN] := p[2 k, wue];
p2m[n_, m_, u_] := (p[n, u] + p[m, u]) / 2;
p3m[n_, m_, s_, u_] := (p[n, u] + p[m, u] + p[s, u]) / 3;
p4m[n_, m_, s_, t_, u_] := (p[n, u] + p[m, u] + p[s, u] + p[t, u]) / 4;
p5m[n_, m_, s_, t_, q_, u_] :=
 (p[n, u] + p[m, u] + p[s, u] + p[t, u] + p[q, u]) / 5;
lin2m[t_, n_, m_, u_] := Line[{p2m[n, m, u], p[t, u]}];
```

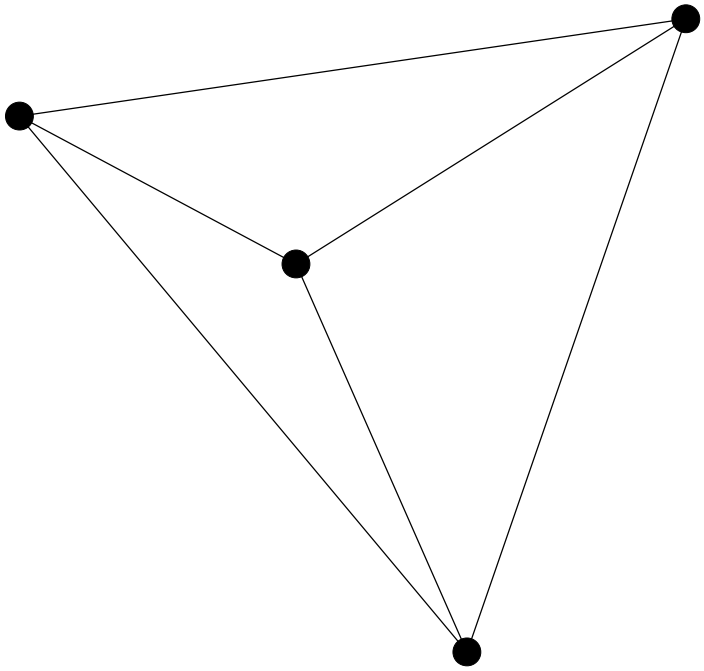
In[75]:=
```
pointT = Prepend[Table[Point[p[k, tet]], {k, 1, 4}], PointSize[0.04]]
```

Out[75]=
```
{PointSize[0.04], Point[{-1, 1, -1}],
 Point[{1, -1, -1}], Point[{1, 1, 1}], Point[{-1, -1, 1}]}
```

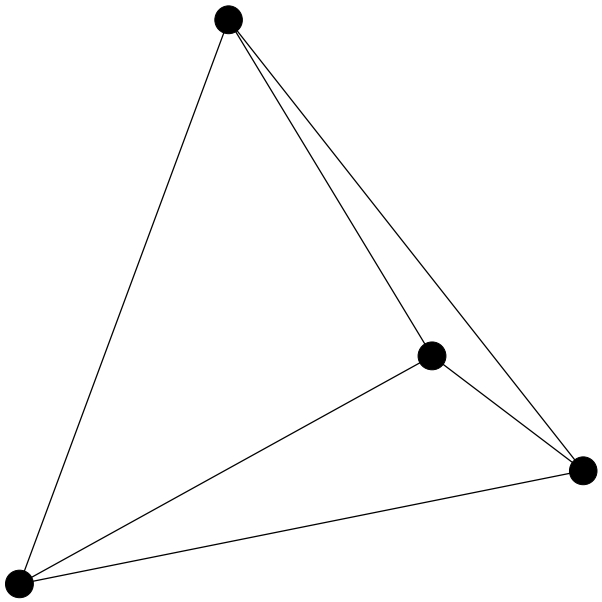
In[76]:=
```
pointTNeg = Prepend[Table[Point[-p[k, tet]], {k, 1, 4}], PointSize[0.04]]
```

Out[76]=
```
{PointSize[0.04], Point[{1, -1, 1}],
 Point[{-1, 1, 1}], Point[{-1, -1, -1}], Point[{1, 1, -1}]}
```

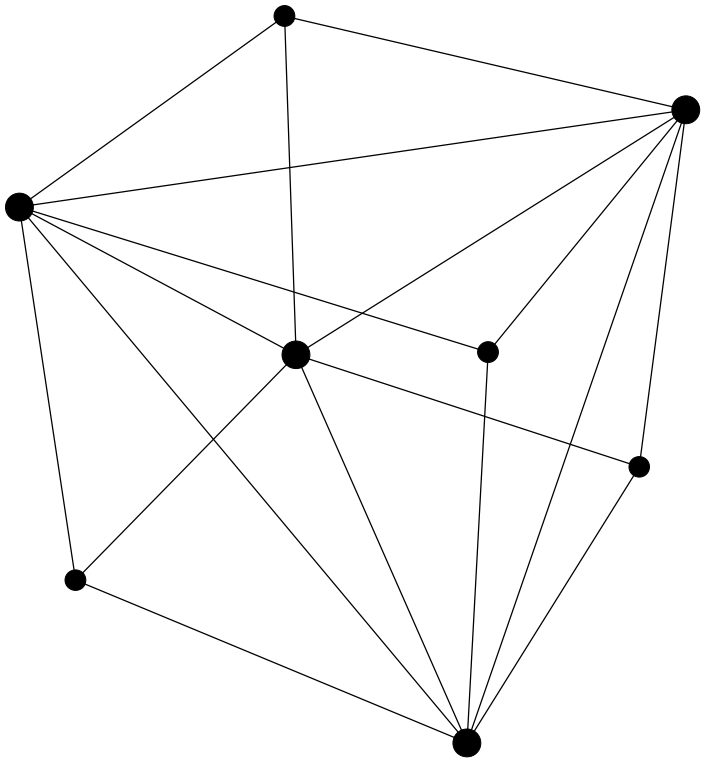
In[77]:=
```
tetraeder1 =
  Show[Graphics3D[{pointT, lin5[1, 2, 3, 4, 1, tet], lin2[1, 3, tet],
     lin2[2, 4, tet]}, Axes → False, AxesEdge → None, Boxed → False]];
```



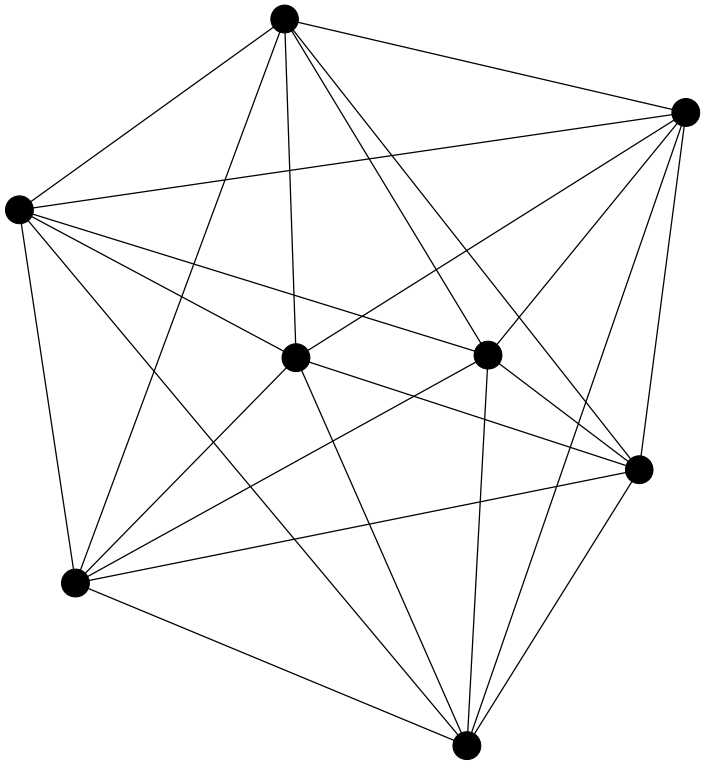
In[78]:=
```
tetraeder2 =
  Show[Graphics3D[{pointTNeg, lin5[1, 2, 3, 4, 1, tetN],
     lin2[1, 3, tetN], lin2[2, 4, tetN]}, Axes → False, AxesEdge → None,
    Boxed → False]];
```



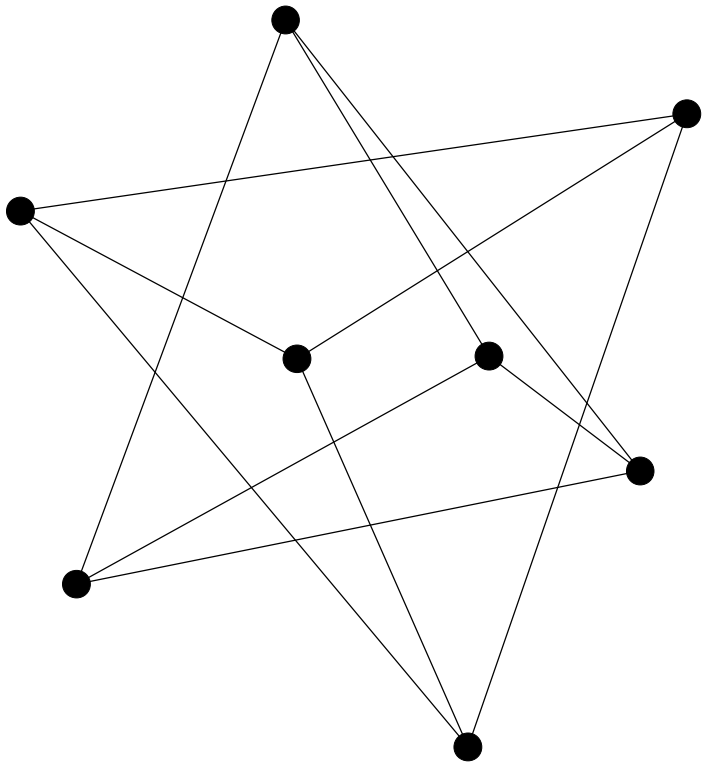
In[79]:=
```
Show[tetraeder1, wuerfel1];
```



In[80]:=
```
Show[tetraeder1, tetraeder2, wuerfel1];
```



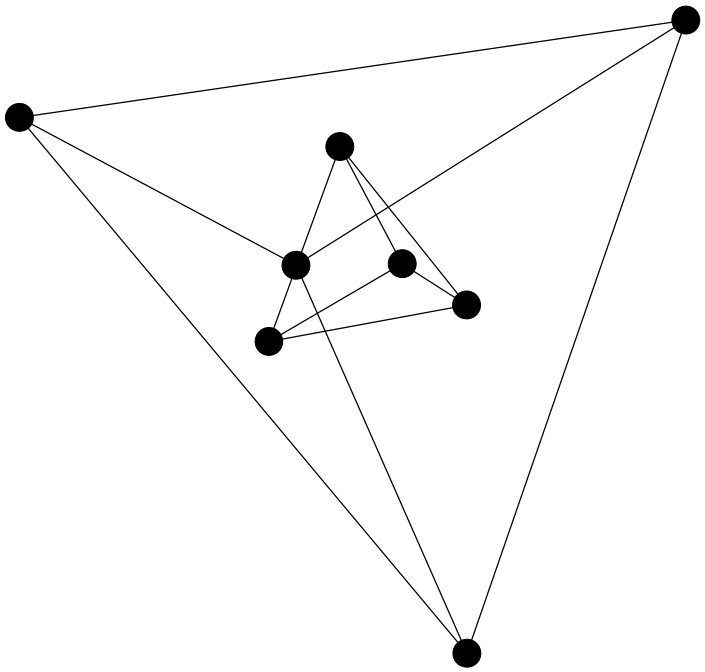
In[81]:=
```
Show[tetraeder1, tetraeder2];
```



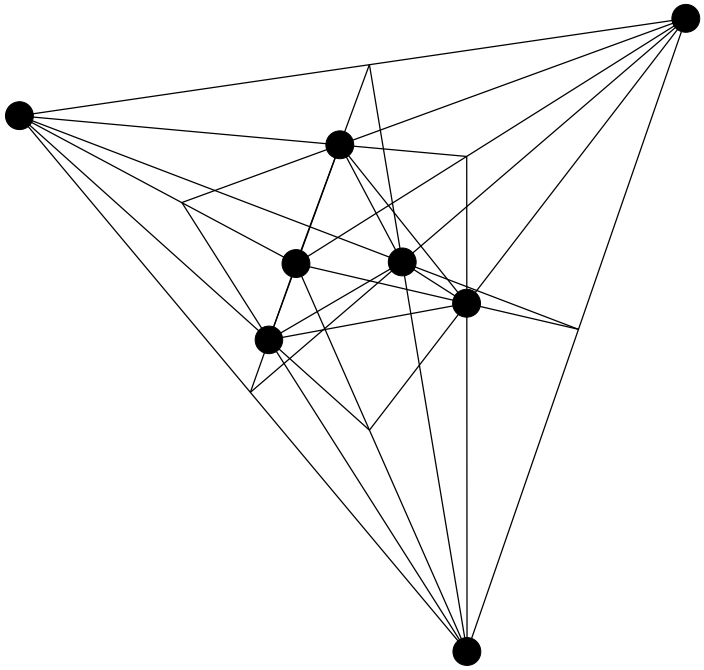
In[82]:=
```
pointTNegIn = Prepend[Table[Point[-p[k, tet] / 3], {k, 1, 4}],
  PointSize[0.04]]
```

Out[82]=
$$\left\{\text{PointSize}[0.04], \text{Point}\left[\left\{\frac{1}{3}, -\frac{1}{3}, \frac{1}{3}\right\}\right], \text{Point}\left[\left\{-\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}\right], \text{Point}\left[\left\{-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}\right\}\right], \text{Point}\left[\left\{\frac{1}{3}, \frac{1}{3}, -\frac{1}{3}\right\}\right]\right\}$$

In[83]:=
```
tetraeder3 =
  Show[Graphics3D[{pointT, pointTNegIn, lin5[1, 2, 3, 4, 1, tet],
     lin2[1, 3, tet], lin2[2, 4, tet], linz5[1, 2, 3, 4, 1, tetN, 1 / 3],
     linz2[1, 3, tetN, 1 / 3], linz2[2, 4, tetN, 1 / 3]}, Axes → False,
    AxesEdge → None, Boxed → False]];
```



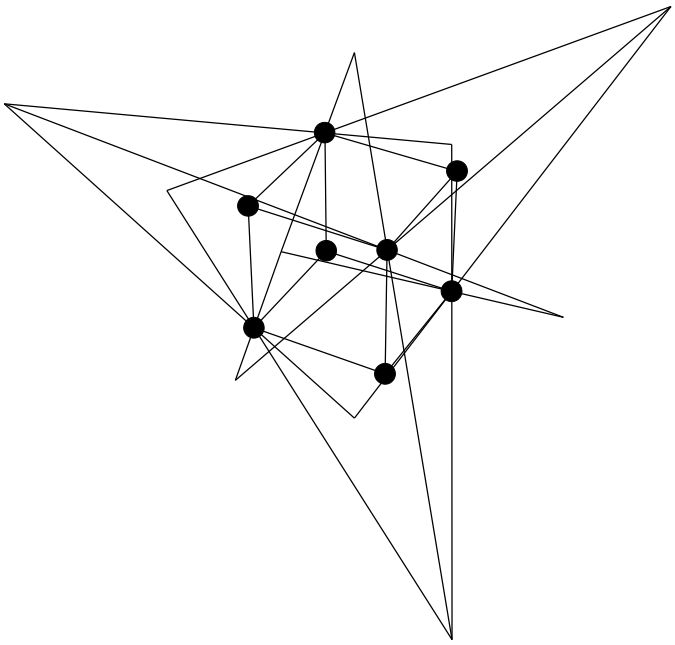
In[84]:=
```
tetraeder4 =
  Show[Graphics3D[{pointT, pointTNegIn, lin5[1, 2, 3, 4, 1, tet],
     lin2[1, 3, tet], lin2[2, 4, tet], linz5[1, 2, 3, 4, 1, tetN, 1 / 3],
     linz2[1, 3, tetN, 1 / 3], linz2[2, 4, tetN, 1 / 3], lin2m[1, 2, 3, tet],
     lin2m[1, 2, 4, tet], lin2m[1, 3, 4, tet], lin2m[2, 1, 3, tet],
     lin2m[2, 1, 4, tet], lin2m[2, 3, 4, tet], lin2m[3, 1, 2, tet],
     lin2m[3, 1, 4, tet], lin2m[3, 2, 4, tet], lin2m[4, 1, 2, tet],
     lin2m[4, 1, 3, tet], lin2m[4, 2, 3, tet]}, Axes → False,
    AxesEdge → None, Boxed → False]];
```



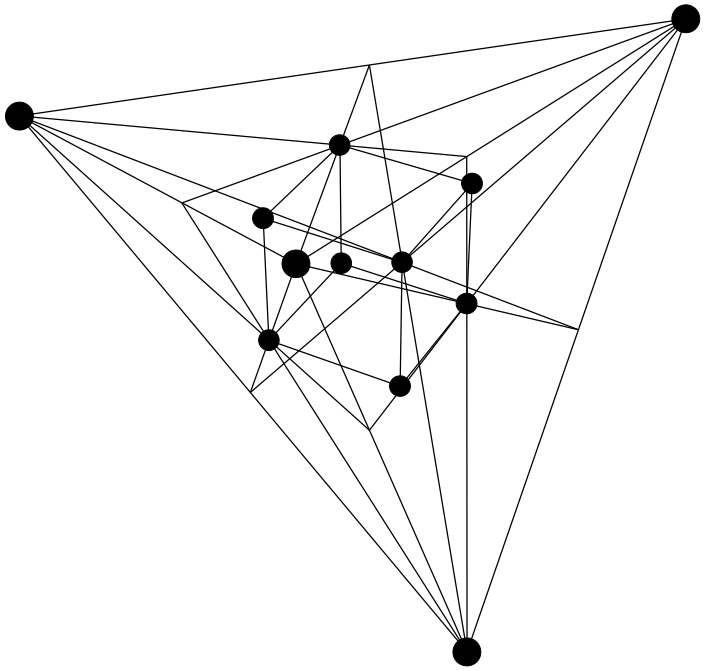
In[85]:=
```
sW = 2;
zW = 1 / (diagVW / 2) * rIT
```

Out[86]=
$$\frac{1}{3}$$

In[87]:=
```
wuerfelT2 =
  Show[Graphics3D[{pointWz[zW], linz5[1, 2, 3, 4, 1, wue, zW],
     linz5[5, 6, 7, 8, 5, wue, zW], linz2[1, 5, wue, zW],
     linz2[2, 6, wue, zW], linz2[3, 7, wue, zW], linz2[4, 8, wue, zW],
     lin2m[1, 2, 3, tet], lin2m[1, 2, 4, tet], lin2m[1, 3, 4, tet],
     lin2m[2, 1, 3, tet], lin2m[2, 1, 4, tet], lin2m[2, 3, 4, tet],
     lin2m[3, 1, 2, tet], lin2m[3, 1, 4, tet], lin2m[3, 2, 4, tet],
     lin2m[4, 1, 2, tet], lin2m[4, 1, 3, tet], lin2m[4, 2, 3, tet]},
    Axes → False, AxesEdge → None, Boxed → False]];
```



In[88]:=
```
Show[tetraeder1, wuerfelT2];
```



## Oktaeder aus dem Würfel heraus

### Berechnungen zum Oktaeder:

In[89]:=
```
sO = Sqrt[2] / 2 sW; diagVO = sW; diagFO = Sqrt[3] / 2 sO;
rUO = sW / 2;  HoeheVO = diagVO / 2 ; volO = 2 sO^2 HoeheVO / 3;
obO = 8 sO diagFO / 2; HoeheFO = sO Sqrt[1^2 - (1 / 2)^2];
rIO = Sqrt[(sO / 2)^2 - ( HoeheFO / 3)^2];
rKO = sO / 2;
```

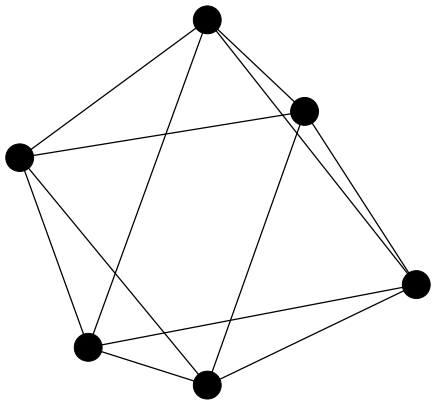
## Graphik zum Oktaeder:

In[91]:=
```
p[1, okt] = p4m[1, 2, 3, 4, wue]; p[2, okt] = p4m[1, 2, 6, 5, wue];
p[3, okt] = p4m[2, 3, 7, 6, wue]; p[4, okt] = p4m[3, 4, 8, 7, wue];
p[5, okt] = p4m[4, 1, 5, 8, wue];
p[6, okt] = p4m[5, 6, 7, 8, wue];
```

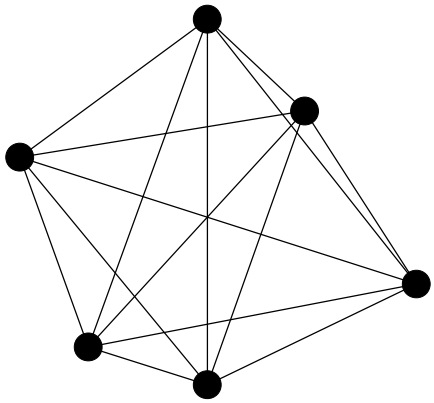
In[92]:=
```
pointO = Prepend[Table[Point[p[k, okt]], {k, 1, 6}], PointSize[0.04]]
```

Out[92]=
```
{PointSize[0.04], Point[{0, 0, -1}], Point[{0, 1, 0}], Point[{-1, 0, 0}],
 Point[{0, -1, 0}], Point[{1, 0, 0}], Point[{0, 0, 1}]}
```

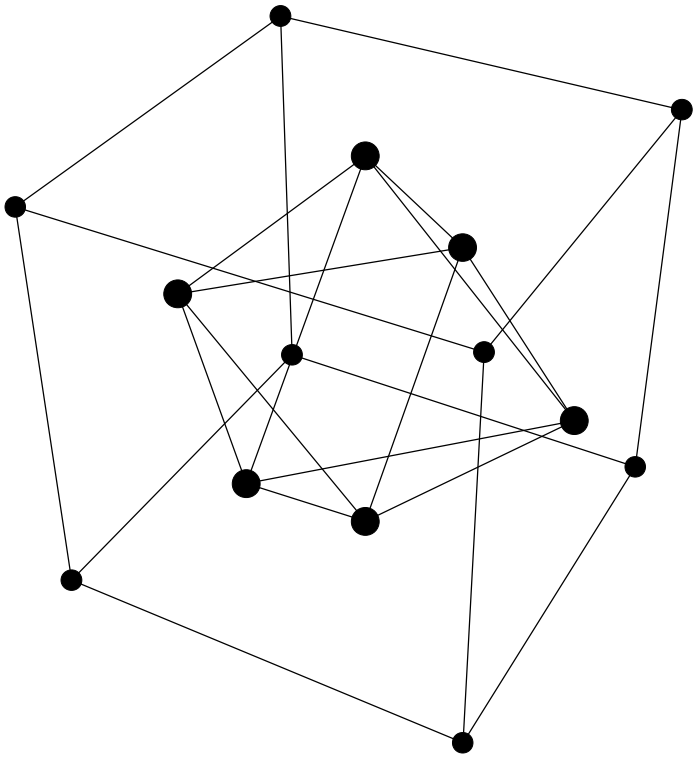
In[93]:=
```
oktaeder1 =
  Show[Graphics3D[{pointO, lin6[1, 2, 3, 4, 5, 6, okt], lin2[1, 3, okt],
     lin2[1, 4, okt], lin2[1, 5, okt], lin2[2, 5, okt], lin2[2, 6, okt],
     lin2[3, 6, okt], lin2[4, 6, okt]}, Axes → False, AxesEdge → None,
    Boxed → False]];
```



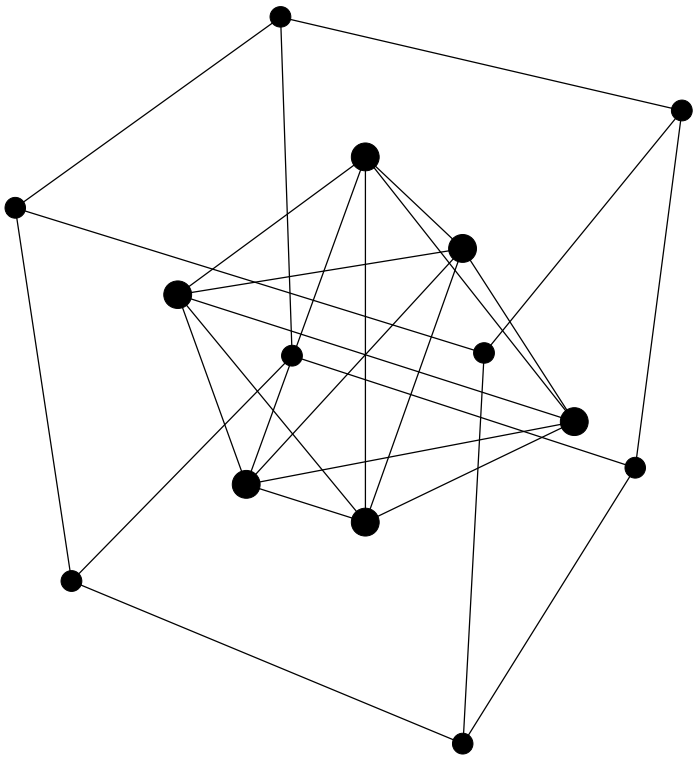
In[94]:=
```
oktaeder2 =
  Show[Graphics3D[{pointO, lin6[1, 2, 3, 4, 5, 6, okt], lin2[1, 3, okt],
     lin2[1, 4, okt], lin2[1, 5, okt], lin2[2, 5, okt], lin2[2, 6, okt],
     lin2[3, 6, okt], lin2[4, 6, okt], lin2[1, 6, okt], lin2[2, 4, okt],
     lin2[3, 5, okt]}, Axes → False, AxesEdge → None, Boxed → False]];
```



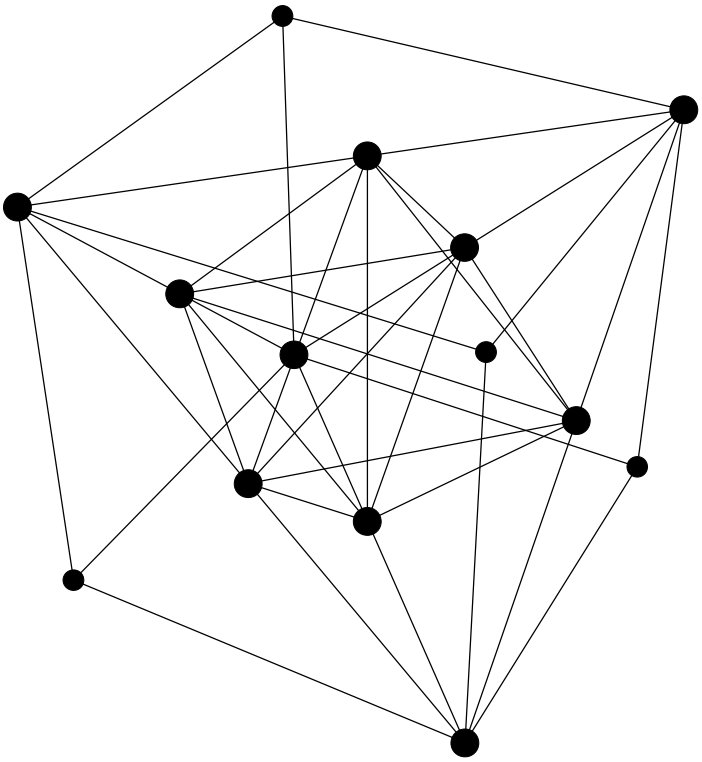
In[95]:=
```
Show[oktaeder1, wuerfel1];
```



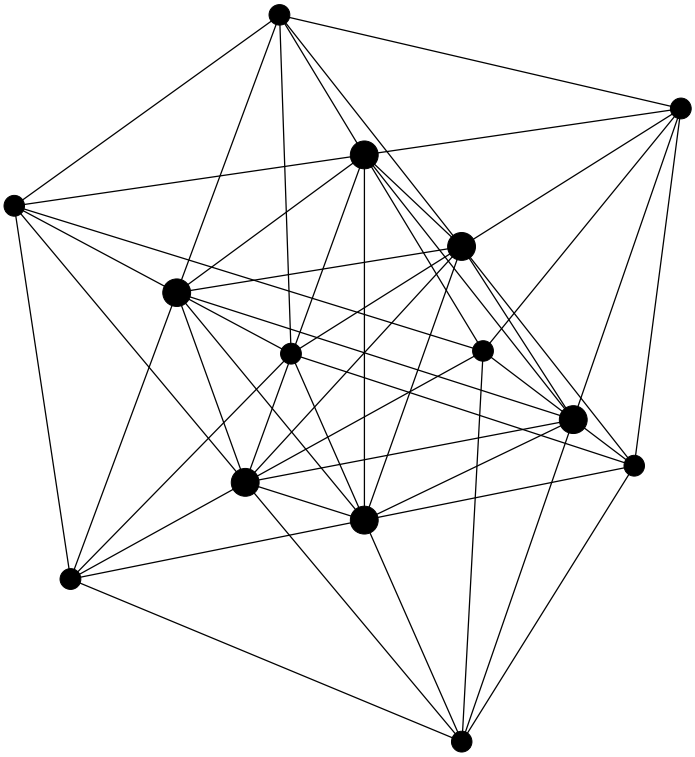
In[96]:=
```
Show[oktaeder2, wuerfel1];
```



In[97]:=
```
Show[oktaeder2, wuerfel1, tetraeder1];
```



In[98]:=
```
Show[oktaeder2, wuerfelF1];
```



In[99]:=
```
sW = 2;
zW = 1 / (diagVW / 2) * rIO
```

Out[100]=
$$\frac{1}{3}$$

In[101]:=
```
HoeheVO
```

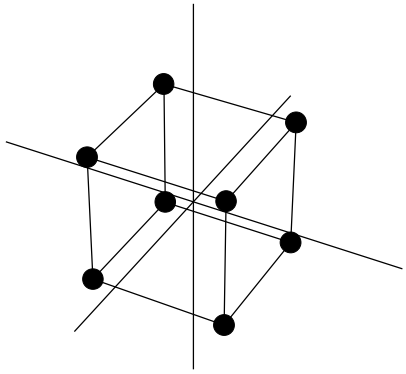
Out[101]=
```
1
```

In[102]:=
```
rIO
```

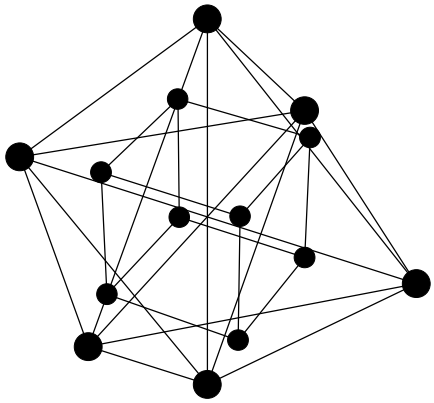
Out[102]=
$$\frac{1}{\sqrt{3}}$$

In[103]:=
```
pmm[1] = p4m[1, 2, 3, 4, wue]; pmm[3] = p4m[1, 2, 5, 6, wue];
pmm[5] = p4m[2, 3, 6, 7, wue]; pmm[4] = p4m[3, 4, 7, 8, wue];
pmm[6] = p4m[4, 1, 8, 5, wue]; pmm[2] = p4m[5, 6, 7, 8, wue];
lin3mm[n_, m_] := Line[{pmm[n], pmm[m]}];
```

In[105]:=
```
wuerfelO2 =
  Show[Graphics3D[{pointWz[zW], linz5[1, 2, 3, 4, 1, wue, zW],
     linz5[5, 6, 7, 8, 5, wue, zW], linz2[1, 5, wue, zW],
     linz2[2, 6, wue, zW], linz2[3, 7, wue, zW], linz2[4, 8, wue, zW],
     lin3mm[1, 2], lin3mm[3, 4], lin3mm[5, 6]}, Axes → False,
    AxesEdge → None, Boxed → False]];
```



In[106]:=
```
Show[oktaeder2, wuerfel02];
```



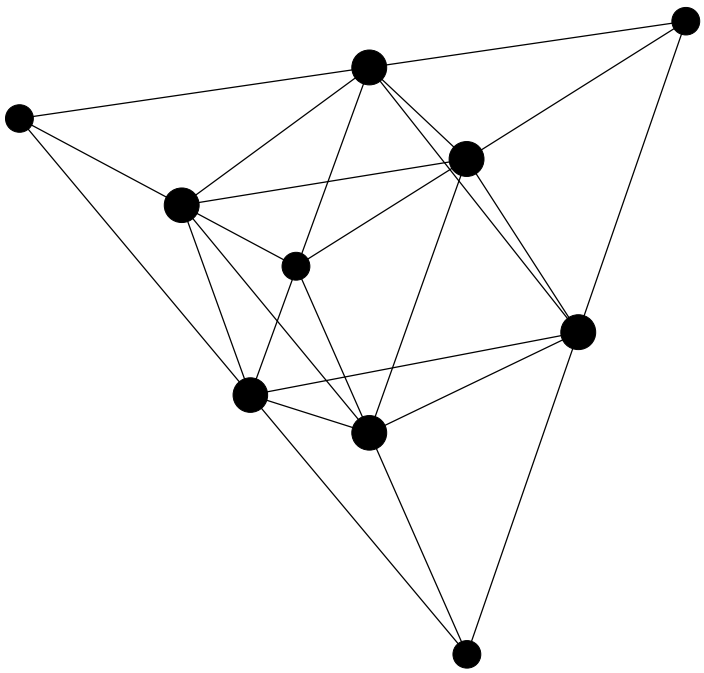
## Oktaeder aus dem Tetraeder heraus

In[107]:=
```
lin2mT[n_, m_] := Line[{pTm[n], pTm[m]}];
```

In[108]:=
```
pTm[1] = p2m[1, 2, tet]; pTm[2] = p2m[1, 3, tet]; pTm[3] = p2m[1, 4, tet];
pTm[4] = p2m[2, 3, tet]; pTm[5] = p2m[2, 4, tet];
pTm[6] = p2m[3, 4, tet];
```

In[109]:=
```
pointTm = Prepend[Table[Point[pTm[k]], {k, 1, 6}], PointSize[0.05]];
```

In[110]:=
```
tetraeder1 =
  Show[Graphics3D[{pointT, lin5[1, 2, 3, 4, 1, tet], lin2[1, 3, tet],
     lin2[2, 4, tet], pointTm,
     lin2mT[1, 2], lin2mT[1, 3], lin2mT[1, 4], lin2mT[1, 5],
     lin2mT[2, 3], lin2mT[2, 4], lin2mT[3, 5], lin2mT[4, 5],
     lin2mT[2, 6], lin2mT[3, 6], lin2mT[4, 6], lin2mT[5, 6]},
    Axes → False, AxesEdge → None, Boxed → False]];
```



## Dodekaeder aus dem Würfel heraus

In[111]:=

```
sD = sW (Sqrt[5] - 1) / 2;
(*Dodekaederseite, Dodekaedersehne = Würfelseite sW   *)
Print["sW = ", sW // N];

s5Dod = sD; (*Dodekaederseite, Fünfeckseite*)
Print["s5Dod=sD = ", sD // N];

diag2Dod = s5Dod (1 + Sqrt[5]) / 2;  (*Dodekaedersehne = Würfelseite sW*)
Print["diag2Dod = ", diag2Dod // N];

s6Dod = s5Dod / Sqrt[1 + ((Sqrt[5] - 1) / 2)^2];
(*Sechseckseite, Fünfeckseite*)
Print["s6Dod = ", s6Dod // N];

rFDod = s6Dod; (*Dodekaederumkreisradius, Sechseckseite   *)
Print["rFDod = ", rFDod // N];

Hs6Dod = Sqrt[s6Dod^2 - (s5Dod / 2)^2];
(*Höhe Fünfeckseite bis Mittelpunkt   *)
Print["Hs6Dod = ", Hs6Dod // N];

Hs2Dod = Sqrt[s5Dod ^2 - (diag2Dod / 2)^2];
(*Höhe Dodekaedersehne bis Ecke*)
Print["Hs2Dod = ", Hs2Dod // N];

dKipp = s6Dod - Hs2Dod;
(*Kippdistanz zwischen Würfelkante und Fünfeckmitte Dodekaeder*)
Print["dKipp = ", dKipp // N];

φ1 = ArcSin[dKipp / (diagFW / 2)] // Simplify // N;
φ = ArcSin[dKipp / (diagFW / 2)];
(*Kippwinkel zwischen Würfel- und Dodekaederachse*)
Print["φ Winkeldifferenz = ", (φ1 / (2 Pi) 360.) // N, " Grad"];

φ2 = -Pi / 4 + φ;
Print["φ2 Kipp = ", (φ2 / (2 Pi) 360.) // N, " Grad"];

(*Halbe Flächendiagonale Würfel*)
Print["Halbe Flächendiagonale Würfel = ", diagFW / 2 // N];
```

```
sW = 2.
```

```
s5Dod=sD = 1.23607
```

```
diag2Dod = 2.
s6Dod = 1.05146
rFDod = 1.05146
Hs6Dod = 0.850651
Hs2Dod = 0.726543
dKipp = 0.32492
φ Winkeldifferenz = 13.2825 Grad
φ2 Kipp = -31.7175 Grad
Halbe Flächendiagonale Würfel = 1.41421
```

In[127]:=
```
N[%]
```

In[128]:=
```
{{1, 0, 0}, {0, Cos[- φ2], -Sin[- φ2]}, {0, Sin[- φ2], Cos[- φ2]}} // N //
  MatrixForm;
```

In[129]:=
```
{{Cos[α], -Sin[α], 0}, {Sin[α], Cos[α], 0}, {0, 0, 1}} // N // MatrixForm;
```

In[130]:=
```
pKipp[k_, α_, wue] :=
  {{Cos[α], -Sin[α], 0}, {Sin[α], Cos[α], 0}, {0, 0, 1}}.
   (({{1, 0, 0}, {0, Cos[-φ2], -Sin[-φ2]}, {0, Sin[-φ2], Cos[-φ2]}}.
     p[k, wue]);
```

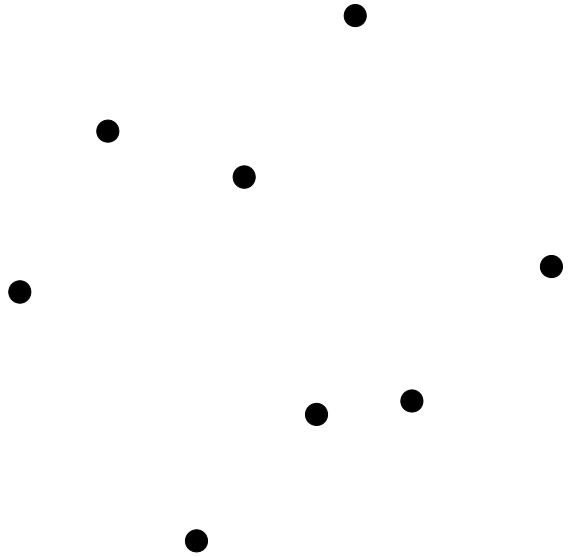
In[131]:=
```
p[1, wue];
```

In[132]:=
```
pointWKipp[α_] :=
  Prepend[Table[Point[pKipp[k, α, wue]], {k, 1, Length[taW]}],
   PointSize[0.03]];
```

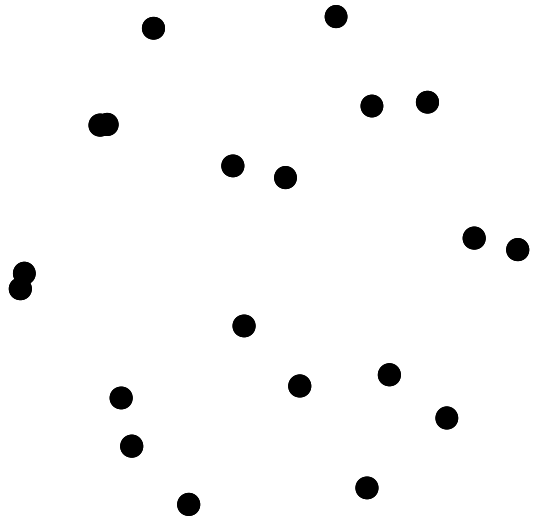
In[133]:=
```
pointWKipp[0] // N;
```

In[134]:=
```
pointD = Table[pointWKipp[k 2 Pi / 5], {k, 1, 1}] // N;
```

In[135]:=
```
Show[Graphics3D[{pointD}, Axes → False, AxesEdge → None, Boxed → False]];
```



In[136]:=
```
pointD = Table[pointWKipp[α 2 Pi / 5], {α, 1, 5}];
Show[Graphics3D[{pointD}, Axes → False, AxesEdge → None, Boxed → False]];
```



In[137]:=
```
? Round
```

```
Round[x] gives the integer closest to x. Mehr…
```

In[138]:=
```
pointWKipp1[α_] :=
 Prepend[Table[Point[Round[10.^6 pKipp[k, α, wue]] / (10.^6)],
   {k, 1, Length[taW]}], PointSize[0.03]]
```

In[139]:=
```
pointD1 = Union[Flatten[Table[pointWKipp1[k 2 Pi / 5], {k, 1, 5}]]];
```

In[140]:=
```
Length[pointD1]
```

Out[140]=
```
21
```

In[141]:=
```
pointNumD[k_] := pointD1[[k]][[1]];
Table[Print["Punkt ", k, " = ", pointNumD[k]], {k, 1, 20}];
pointD2 = Table[pointNumD[k], {k, 1, 20}];
```

```
Punkt 1 = {-1.61803, -0.525731, -0.32492}
Punkt 2 = {-1.61803, 0.525731, 0.32492}
Punkt 3 = {-1., -1.37638, 0.32492}
Punkt 4 = {-1., -0.32492, -1.37638}
Punkt 5 = {-1., 0.32492, 1.37638}
Punkt 6 = {-1., 1.37638, -0.32492}
Punkt 7 = {-0.618034, -0.850651, 1.37638}
Punkt 8 = {-0.618034, 0.850651, -1.37638}
Punkt 9 = {0, -1.7013, -0.32492}
Punkt 10 = {0, -1.05146, -1.37638}
Punkt 11 = {0, 1.05146, 1.37638}
Punkt 12 = {0, 1.7013, 0.32492}
Punkt 13 = {0.618034, -0.850651, 1.37638}
Punkt 14 = {0.618034, 0.850651, -1.37638}
Punkt 15 = {1., -1.37638, 0.32492}
Punkt 16 = {1., -0.32492, -1.37638}
Punkt 17 = {1., 0.32492, 1.37638}
Punkt 18 = {1., 1.37638, -0.32492}
Punkt 19 = {1.61803, -0.525731, -0.32492}
Punkt 20 = {1.61803, 0.525731, 0.32492}
```

In[142]:=
```
TabPointNumbD2 = Table[ pointNumD[k], {k, 1, 20}];
TabPointNumbD2 // MatrixForm
```

Out[142]//MatrixForm=

| | | |
|---|---|---|
| −1.61803 | −0.525731 | −0.32492 |
| −1.61803 | 0.525731 | 0.32492 |
| −1. | −1.37638 | 0.32492 |
| −1. | −0.32492 | −1.37638 |
| −1. | 0.32492 | 1.37638 |
| −1. | 1.37638 | −0.32492 |
| −0.618034 | −0.850651 | 1.37638 |
| −0.618034 | 0.850651 | −1.37638 |
| 0 | −1.7013 | −0.32492 |
| 0 | −1.05146 | −1.37638 |
| 0 | 1.05146 | 1.37638 |
| 0 | 1.7013 | 0.32492 |
| 0.618034 | −0.850651 | 1.37638 |
| 0.618034 | 0.850651 | −1.37638 |
| 1. | −1.37638 | 0.32492 |
| 1. | −0.32492 | −1.37638 |
| 1. | 0.32492 | 1.37638 |
| 1. | 1.37638 | −0.32492 |
| 1.61803 | −0.525731 | −0.32492 |
| 1.61803 | 0.525731 | 0.32492 |

In[143]:=
```
? Select
```

```
Select[list, crit] picks out all elements ei of list
   for which crit[ei] is True. Select[list, crit, n] picks
   out the first n elements for which crit[ei] is True. Mehr…
```

In[144]:=
```
DodTest = Select[TabPointNumbD2, (#[[1]] > 0  && #[[2]] > 0) &]
```

Out[144]=
```
{{0.618034, 0.850651, −1.37638}, {1., 0.32492, 1.37638},
 {1., 1.37638, −0.32492}, {1.61803, 0.525731, 0.32492}}
```

In[145]:=
```
DodFuenf[1] = Select[TabPointNumbD2, (#[[3]] > 1) &];
DodFuenf[7] = Select[TabPointNumbD2, (#[[3]] < -1) &];
```

In[147]:=
```
DodFuenf[2] = Select[TabPointNumbD2,
   (#[[1]] > 0  && #[[2]] < 1 && #[[3]] > -0.4) &];
DodFuenf[8] = Select[TabPointNumbD2,
   (#[[1]] < 0  && #[[2]] > -1 && #[[3]] < 0.4) &];
```

In[149]:=
```
DodFuenf[3] = Select[TabPointNumbD2,
   (#[[1]] >= 0 && #[[2]] ≥ 0 && #[[3]] > -0.4) &];
DodFuenf[9] = Select[TabPointNumbD2,
   (#[[1]] <= 0 && #[[2]] ≤ 0 && #[[3]] < 0.4) &];
```

In[151]:=
```
DodFuenf[4] = Select[TabPointNumbD2,
   (#[[1]] ≤ 0 && #[[2]] > -0.3 && #[[3]] > -0.4) &];
DodFuenf[10] = Select[TabPointNumbD2,
   (#[[1]] >= 0 && #[[2]] < 0.3 && #[[3]] < 0.4) &];
```

In[153]:=
```
DodFuenf[5] = Select[TabPointNumbD2,
   (#[[1]] < -0.3 && #[[2]] ≤ 0.6 && #[[3]] > -0.4) &];
DodFuenf[11] = Select[TabPointNumbD2,
   (#[[1]] > 0.3 && #[[2]] ≥ -0.6 && #[[3]] < 0.4) &];
```

In[155]:=
```
DodFuenf[6] = Select[TabPointNumbD2, (#[[2]] < -0.7 && #[[3]] > -0.4) &];
DodFuenf[12] = Select[TabPointNumbD2, ( #[[2]] > 0.7 && #[[3]] < 0.4) &];
```

In[157]:=
```
prnt[k_] := Table[pDodNum[k, n] = DodFuenf[k][[n]];
  Print["Punkt[", k, ",", n, "] =  ", DodFuenf[k][[n]]], {n, 1, 5}]
```

In[158]:=
```
DodFuenf[1][[5]];
```

In[159]:=
```
DodWowrk[u_] := Table[{DodFuenf[u][[k]], DodFuenf[u][[m]]},
   {k, 1, 5}, {m, 1, 5}];
```

In[160]:=
```
DodWowrkTest1 =
  Flatten[Table[{DodFuenf[1][[k]], DodFuenf[1][[m]]}, {k, 1, 5},
    {m, 1, 5}], 1];
```

In[161]:=
```
vLen[DodWowrkTest1[[5]][[1]] - DodWowrkTest1[[5]][[2]]];
```

In[162]:=
```
vLen[u_] := Sqrt[u.u];
```

In[163]:=
```
Union[
  Flatten[
   Table[
    If[
     s5Dod - 0.1 < vLen[DodWowrkTest1[[k]][[1]] -
         DodWowrkTest1[[k]][[2]]] &&
      s5Dod + 0.1 > vLen[DodWowrkTest1[[k]][[1]] -
         DodWowrkTest1[[k]][[2]]], DodWowrkTest1[[k]], zZZZ],
    {k, 1, Length[DodWowrkTest1]}], 1]];
```

In[164]:=
```
Table[prnt[k], {k, 1, 12}]
```

```
Punkt[1,1] = {-1., 0.32492, 1.37638}
Punkt[1,2] = {-0.618034, -0.850651, 1.37638}
Punkt[1,3] = {0, 1.05146, 1.37638}
Punkt[1,4] = {0.618034, -0.850651, 1.37638}
Punkt[1,5] = {1., 0.32492, 1.37638}
Punkt[2,1] = {0.618034, -0.850651, 1.37638}
Punkt[2,2] = {1., -1.37638, 0.32492}
Punkt[2,3] = {1., 0.32492, 1.37638}
Punkt[2,4] = {1.61803, -0.525731, -0.32492}
Punkt[2,5] = {1.61803, 0.525731, 0.32492}
Punkt[3,1] = {0, 1.05146, 1.37638}
Punkt[3,2] = {0, 1.7013, 0.32492}
Punkt[3,3] = {1., 0.32492, 1.37638}
Punkt[3,4] = {1., 1.37638, -0.32492}
Punkt[3,5] = {1.61803, 0.525731, 0.32492}
Punkt[4,1] = {-1.61803, 0.525731, 0.32492}
Punkt[4,2] = {-1., 0.32492, 1.37638}
Punkt[4,3] = {-1., 1.37638, -0.32492}
Punkt[4,4] = {0, 1.05146, 1.37638}
Punkt[4,5] = {0, 1.7013, 0.32492}
Punkt[5,1] = {-1.61803, -0.525731, -0.32492}
Punkt[5,2] = {-1.61803, 0.525731, 0.32492}
Punkt[5,3] = {-1., -1.37638, 0.32492}
Punkt[5,4] = {-1., 0.32492, 1.37638}
Punkt[5,5] = {-0.618034, -0.850651, 1.37638}
```

```
Punkt[6,1] = {-1., -1.37638, 0.32492}
Punkt[6,2] = {-0.618034, -0.850651, 1.37638}
Punkt[6,3] = {0, -1.7013, -0.32492}
Punkt[6,4] = {0.618034, -0.850651, 1.37638}
Punkt[6,5] = {1., -1.37638, 0.32492}
Punkt[7,1] = {-1., -0.32492, -1.37638}
Punkt[7,2] = {-0.618034, 0.850651, -1.37638}
Punkt[7,3] = {0, -1.05146, -1.37638}
Punkt[7,4] = {0.618034, 0.850651, -1.37638}
Punkt[7,5] = {1., -0.32492, -1.37638}
Punkt[8,1] = {-1.61803, -0.525731, -0.32492}
Punkt[8,2] = {-1.61803, 0.525731, 0.32492}
Punkt[8,3] = {-1., -0.32492, -1.37638}
Punkt[8,4] = {-1., 1.37638, -0.32492}
Punkt[8,5] = {-0.618034, 0.850651, -1.37638}
Punkt[9,1] = {-1.61803, -0.525731, -0.32492}
Punkt[9,2] = {-1., -1.37638, 0.32492}
Punkt[9,3] = {-1., -0.32492, -1.37638}
Punkt[9,4] = {0, -1.7013, -0.32492}
Punkt[9,5] = {0, -1.05146, -1.37638}
Punkt[10,1] = {0, -1.7013, -0.32492}
Punkt[10,2] = {0, -1.05146, -1.37638}
Punkt[10,3] = {1., -1.37638, 0.32492}
Punkt[10,4] = {1., -0.32492, -1.37638}
Punkt[10,5] = {1.61803, -0.525731, -0.32492}
Punkt[11,1] = {0.618034, 0.850651, -1.37638}
Punkt[11,2] = {1., -0.32492, -1.37638}
Punkt[11,3] = {1., 1.37638, -0.32492}
Punkt[11,4] = {1.61803, -0.525731, -0.32492}
Punkt[11,5] = {1.61803, 0.525731, 0.32492}
Punkt[12,1] = {-1., 1.37638, -0.32492}
Punkt[12,2] = {-0.618034, 0.850651, -1.37638}
Punkt[12,3] = {0, 1.7013, 0.32492}
Punkt[12,4] = {0.618034, 0.850651, -1.37638}
```

```
Punkt[12,5] = {1., 1.37638, -0.32492}
```

Out[164]=

```
{{Null, Null, Null, Null, Null}, {Null, Null, Null, Null, Null},
 {Null, Null, Null, Null, Null}, {Null, Null, Null, Null, Null},
 {Null, Null, Null, Null, Null}, {Null, Null, Null, Null, Null},
 {Null, Null, Null, Null, Null}, {Null, Null, Null, Null, Null},
 {Null, Null, Null, Null, Null}, {Null, Null, Null, Null, Null},
 {Null, Null, Null, Null, Null}, {Null, Null, Null, Null, Null}}
```

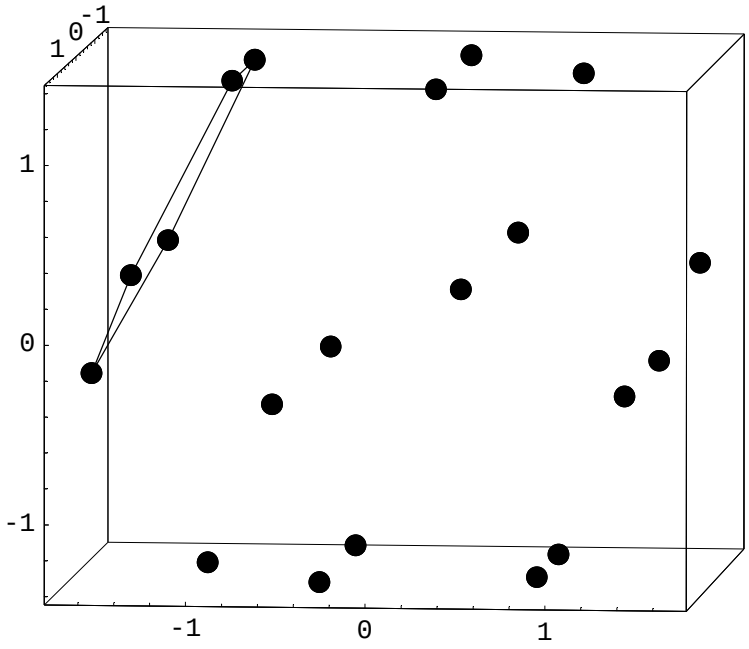
In[165]:=

```
kk = 1;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 7;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 2;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 8;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 3;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 9;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 4;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 10;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 6;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
kk = 12;
DodFuenfNeu[kk] = {pDodNum[kk, 1], pDodNum[kk, 2], pDodNum[kk, 4],
  pDodNum[kk, 5], pDodNum[kk, 3], pDodNum[kk, 1]};
```

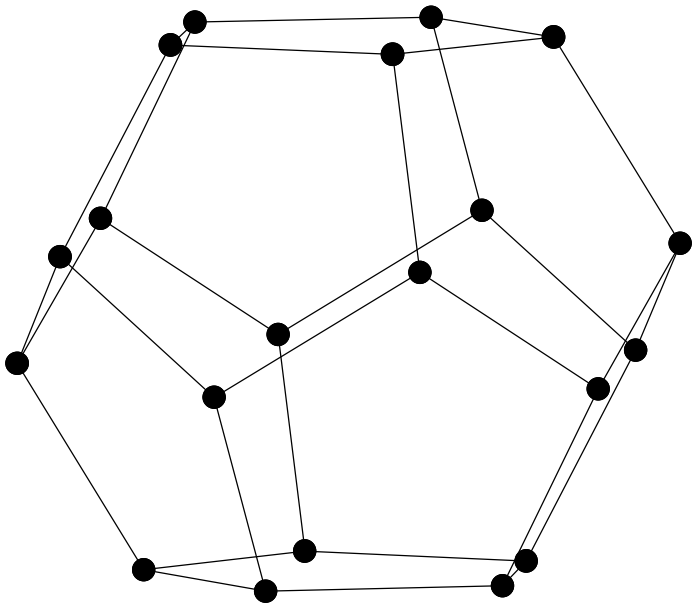
In[175]:=

```
lDod[k_] := Line[DodFuenfNeu[k]]; lDodAll = Table[lDod[k], {k, 1, 12}];

sehnenDodAll = Table[lDod[k], {k, 1, 12}];
```

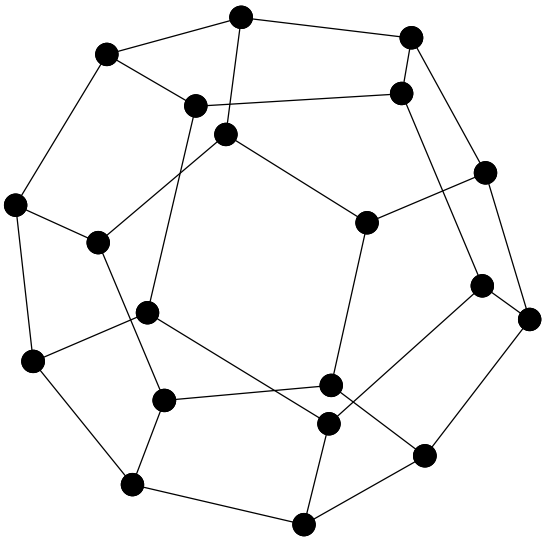
In[177]:=
```
pointDLTestEinzeln[k_] :=
  Show[Graphics3D[{pointD, (*Line[{pDodNum[1,1],pDodNum[1,2]}]*)
     lDod[k]}], Axes → True, ViewPoint → {100, 10, 10}];

pointDLTestEinzeln[6];
```




In[179]:=
```
pointDL =
  Show[Graphics3D[{pointD, Line[{pDodNum[1, 1], pDodNum[1, 2]}],
     lDod[1], lDod[2], lDod[3], lDod[4], lDod[5], lDod[6],
     lDod[7], lDod[8], lDod[9], lDod[10], lDod[11], lDod[12]},
    Axes → False, AxesEdge → None, Boxed → False],
   ViewPoint → {100, 10, 10}];
```

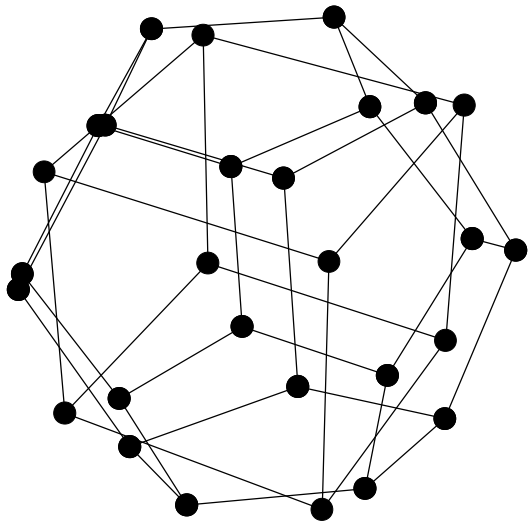


In[180]:=
```
pointDL =
  Show[Graphics3D[{pointD, Line[{pDodNum[1, 1], pDodNum[1, 2]}],
     lDod[1], lDod[2], lDod[3], lDod[4], lDod[5], lDod[6],
     lDod[7], lDod[8], lDod[9], lDod[10], lDod[11], lDod[12]},
    Axes → False, AxesEdge → None, Boxed → False],
   ViewPoint -> {2.726, -1.531, 1.294}];
```

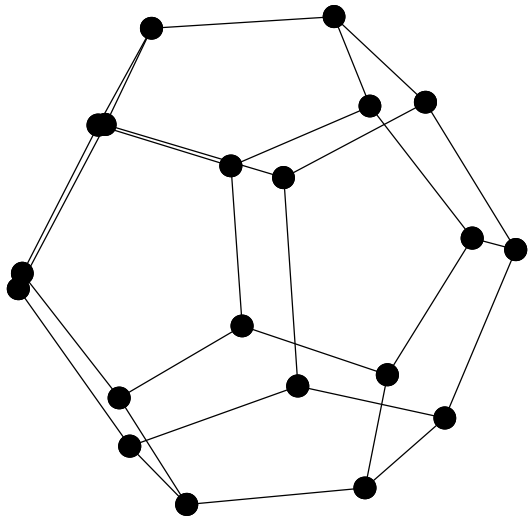


In[181]:=
```
nimm2 = pointDL;
```

In[182]:=
```
zeig1 = Show[nimm1, nimm2];
```

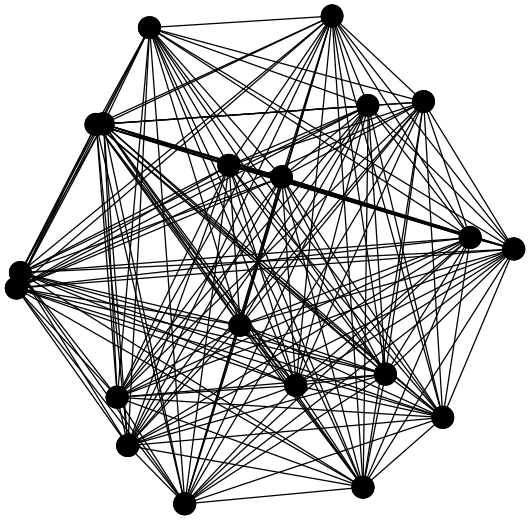


In[183]:=
```
pointDL = Show[Graphics3D[{pointD, lDodAll}, Axes → False,
    AxesEdge → None, Boxed → False]];
```

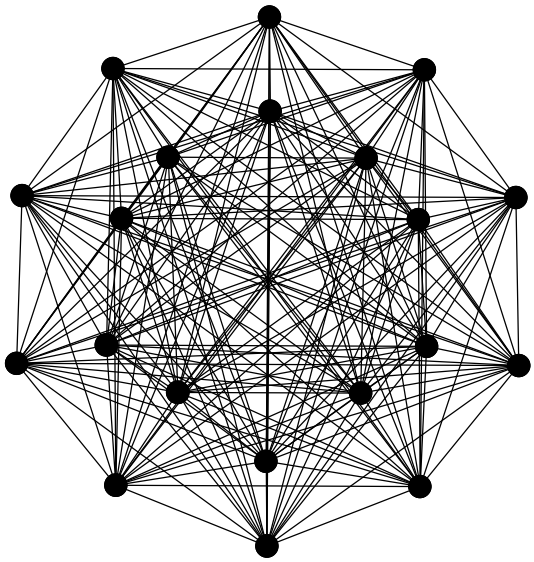


In[184]:=
```
allLinesDod =
  Flatten[
   Table[Flatten[Table[Line[{pDodNum[j, i], pDodNum[k, n]}],
      {j, 1, k}, {i, 1, n}], 1], {k, 1, 12}, {n, 1, 5}]];
```

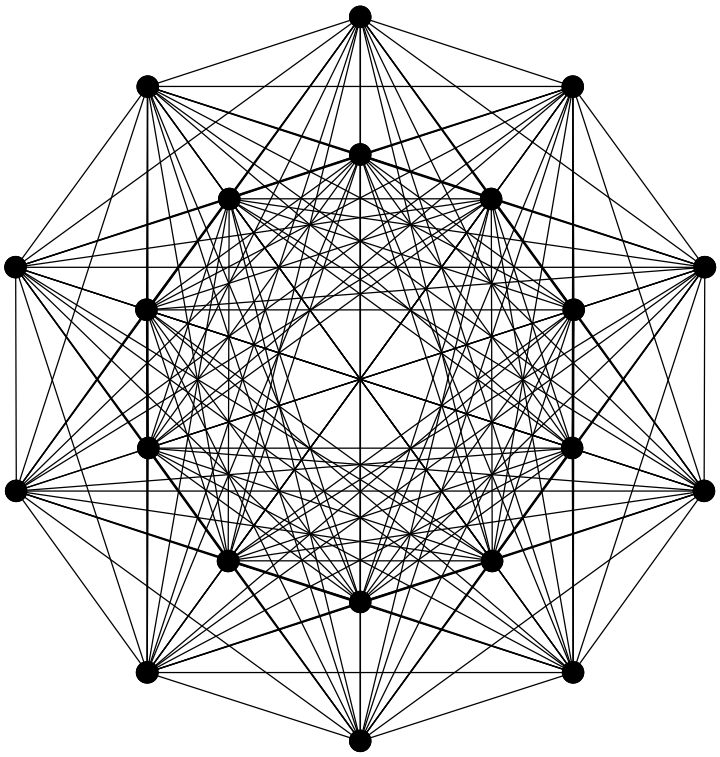
In[185]:=
```
pointDL = Show[Graphics3D[{pointD, allLinesDod}, Axes → False,
    AxesEdge → None, Boxed → False]];
```



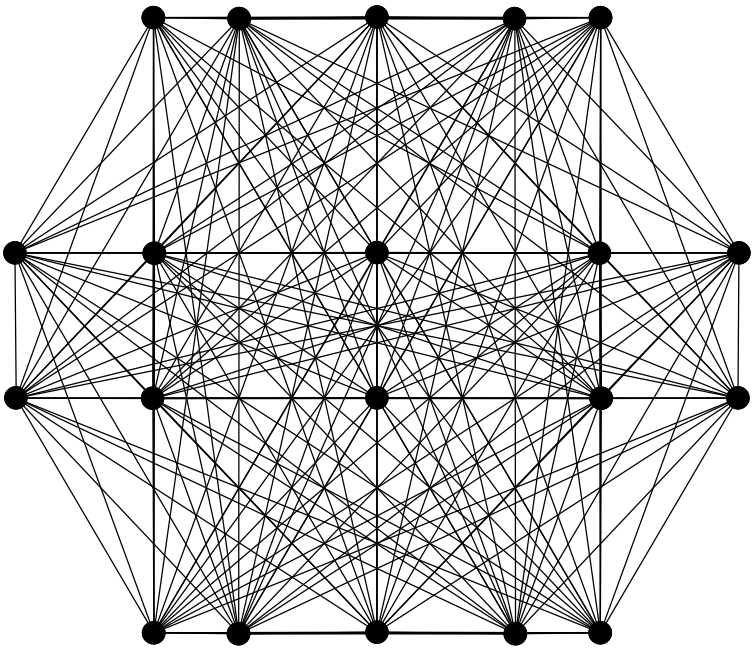
In[186]:=
```
pointDL = Show[Graphics3D[{pointD, allLinesDod}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {5, 7, 5}];
```



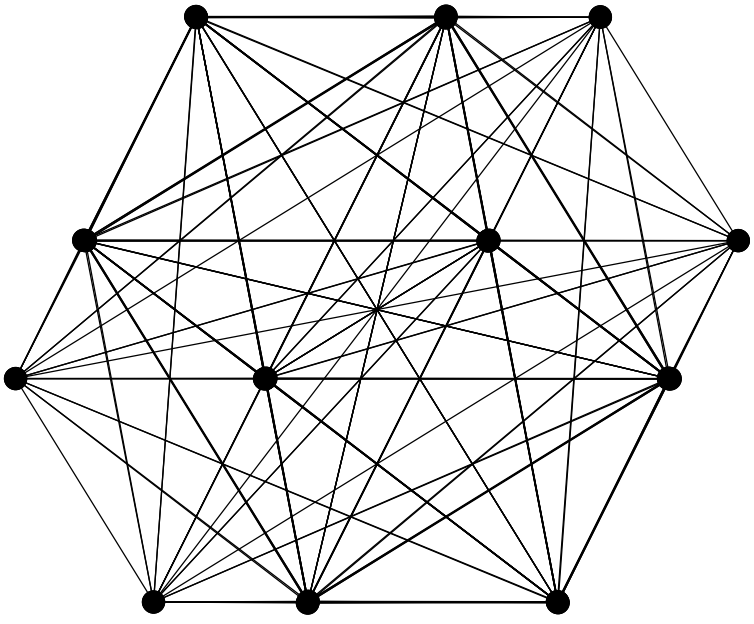
In[187]:=
```
pointDL = Show[Graphics3D[{pointD, allLinesDod}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {0, 0, 100}];
```



In[188]:=
```
pointDL = Show[Graphics3D[{pointD, allLinesDod}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {0, 100, 0}];
```



In[189]:=
```
pointDL = Show[Graphics3D[{pointD, allLinesDod}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {100, 0, 0}];
```



## Kanten und Flächenwinkel am Dodekaeder

In[190]:=
```
TabPointNumbD2 // MatrixForm
```

Out[190]//MatrixForm=

| | | |
|---|---|---|
| -1.61803 | -0.525731 | -0.32492 |
| -1.61803 | 0.525731 | 0.32492 |
| -1. | -1.37638 | 0.32492 |
| -1. | -0.32492 | -1.37638 |
| -1. | 0.32492 | 1.37638 |
| -1. | 1.37638 | -0.32492 |
| -0.618034 | -0.850651 | 1.37638 |
| -0.618034 | 0.850651 | -1.37638 |
| 0 | -1.7013 | -0.32492 |
| 0 | -1.05146 | -1.37638 |
| 0 | 1.05146 | 1.37638 |
| 0 | 1.7013 | 0.32492 |
| 0.618034 | -0.850651 | 1.37638 |
| 0.618034 | 0.850651 | -1.37638 |
| 1. | -1.37638 | 0.32492 |
| 1. | -0.32492 | -1.37638 |
| 1. | 0.32492 | 1.37638 |
| 1. | 1.37638 | -0.32492 |
| 1.61803 | -0.525731 | -0.32492 |
| 1.61803 | 0.525731 | 0.32492 |

In[191]:=
```
vLen[v_] := Sqrt[v.v];
```

In[192]:=
```
angle[v1_, v2_] := ArcCos[(v1.v2) / (vLen[v1] vLen[v2])] 360 / (2 Pi)
```

In[193]:=
```
{pointNumD[5], pointNumD[7], pointNumD[11], pointNumD[13],
  pointNumD[17]} // MatrixForm
```

Out[193]//MatrixForm=

| | | |
|---|---|---|
| -1. | 0.32492 | 1.37638 |
| -0.618034 | -0.850651 | 1.37638 |
| 0 | 1.05146 | 1.37638 |
| 0.618034 | -0.850651 | 1.37638 |
| 1. | 0.32492 | 1.37638 |

In[194]:=
```
{angle[pointNumD[11], {0, 0, 1}], angle[pointNumD[11], pointNumD[12]]}
```

Out[194]=
```
{37.3774, 41.8103}
```

In[195]:=
```
angle[pointNumD[11] - pointNumD[12], {0, 0, 1} - pointNumD[11]]
```

Out[195]=
```
77.9779
```

In[196]:=
```
pointNumD[work] = (pointNumD[7] + pointNumD[13]) / 2
```

Out[196]=
```
{0., -0.850651, 1.37638}
```

In[197]:=
```
{angle[pointNumD[work], {0, 0, 1}],
 angle[pointNumD[work], pointNumD[9]]}
```

Out[197]=
```
{31.7175, 69.0948}
```

In[198]:=
```
angle[pointNumD[9] - pointNumD[work], pointNumD[work] - {0, 0, 1}]
```

Out[198]=
```
87.3026
```

## Ikosaeder aus den Seitenmittelpunkten des Dodekaeders heraus

■ **Achtung: In *Mathematica* Version 5.0 ist die Funktion "Total" im Kern eingebaut !!!!!!**

In[199]:=
```
Total[x_] := Sum[x[[k]], {k, 1, Length[x]}]
```

```
SetDelayed::write : Tag Total in Total[x_] is Protected. Mehr…
```

Out[199]=
```
$Failed
```

■ **Achtung: In *Mathematica* Version 5.0 ist die Funktion "Total" im Kern eingebaut !!!!!!**

In[200]:=
```
Total[{1, 2, 3, 4, 5}]
```

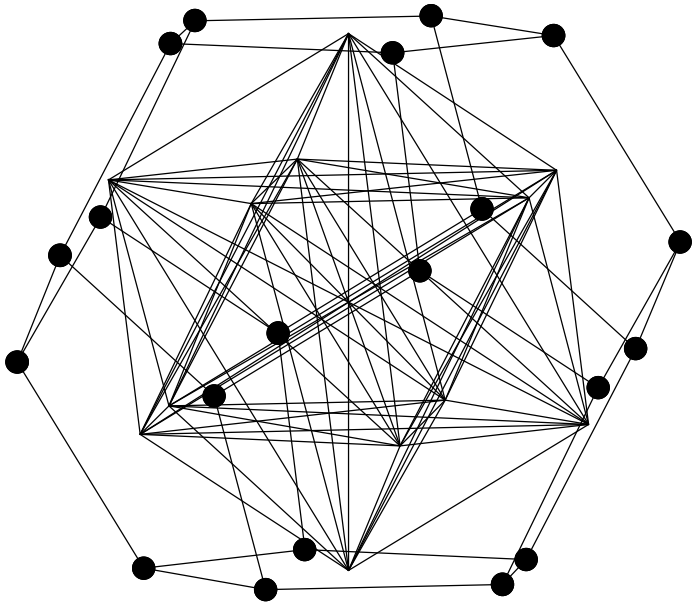
Out[200]=
```
15
```

In[201]:=
```
Total[DodFuenfNeu[7]];
```

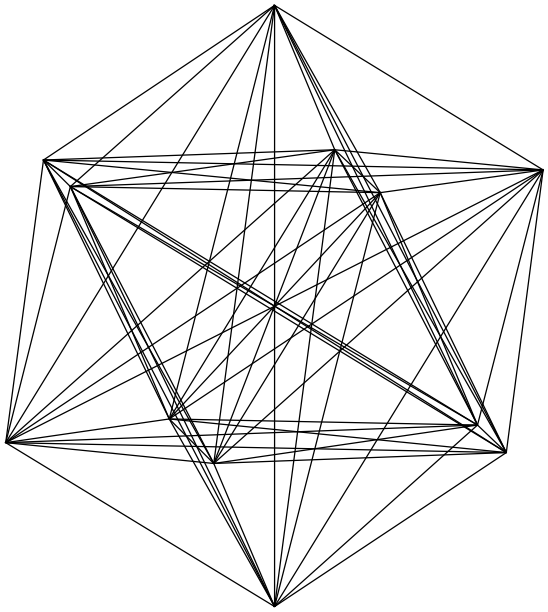
In[202]:=
```
pDodNum = Table[Total[DodFuenf[k]] / 5, {k, 1, 12}];
```

In[203]:=
```
allLinesIkos =
  Flatten[Table[Table[Line[{pDodNum[[j]], pDodNum[[k]]}], {j, 1, k}],
    {k, 1, 12}]];
```

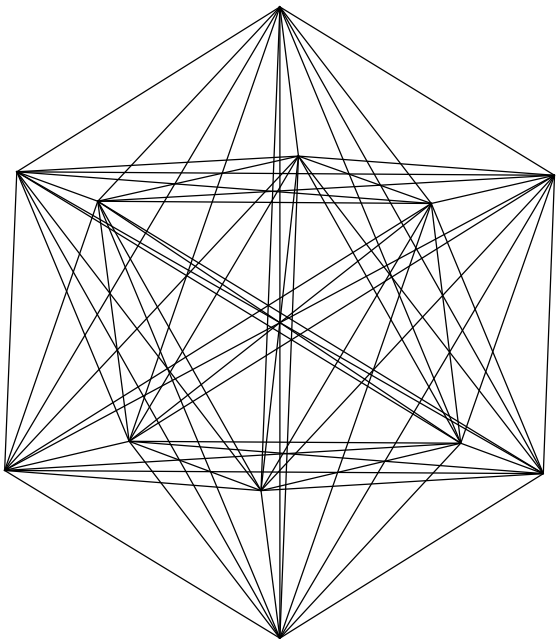
In[204]:=
```
pointIL = Show[Graphics3D[{pointD, allLinesIkos, lDodAll},
    Axes → False, AxesEdge → None, Boxed → False],
   ViewPoint → {100, 10, 10}];
```



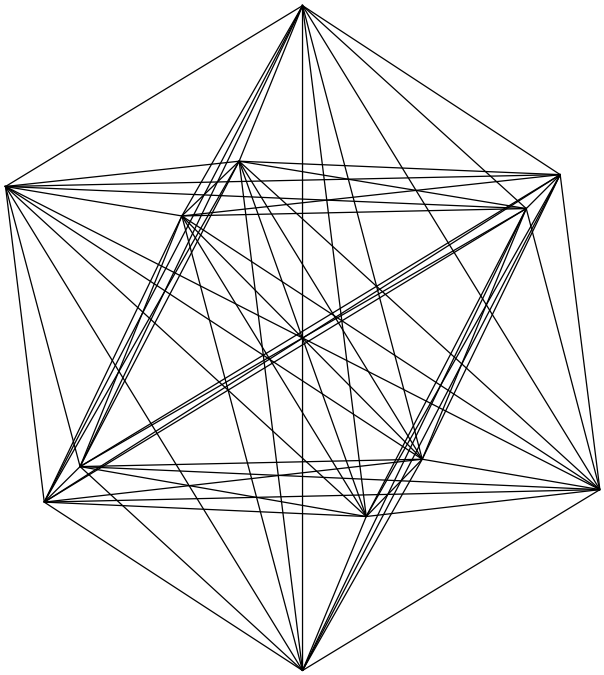
In[205]:=
```
pointIL = Show[Graphics3D[{allLinesIkos}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {100, 60, 10}];
```



In[206]:=
```
pointIL = Show[Graphics3D[{allLinesIkos}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {100, 40, 10}];
```

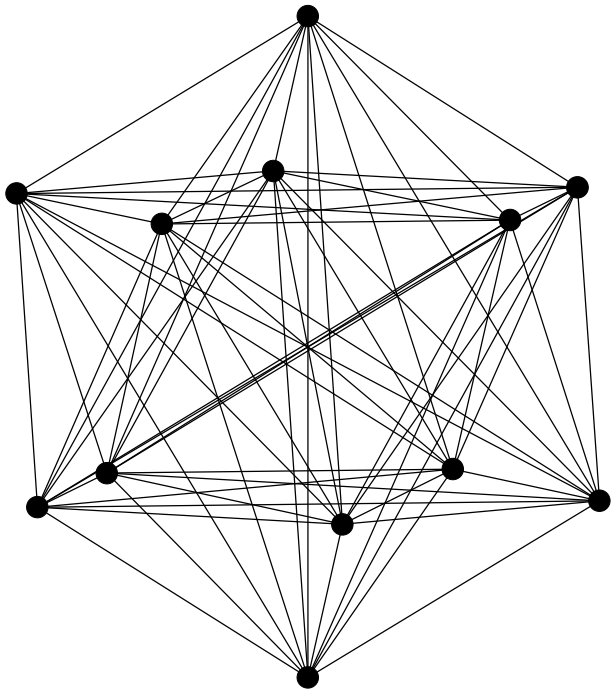


In[207]:=
```
pointIL = Show[Graphics3D[{allLinesIkos}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {100, 10, 10}];
```

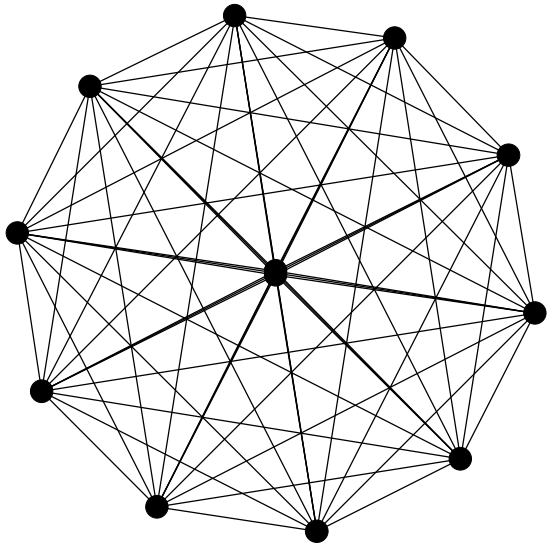


In[208]:=
```
pointsI = Prepend[Map[Point, pDodNum], PointSize[0.03]];
```

In[209]:=
```
pointIL = Show[Graphics3D[{pointsI, allLinesIkos}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {100, 20, 10}];
```



In[210]:=
```
pointIL = Show[Graphics3D[{pointsI, allLinesIkos}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {0.5, 0.5, 100}];
```



In[211]:=
```
allDiffIkos =
  Flatten[Table[Table[pDodNum[[j]] - pDodNum[[k]], {j, 1, k}],
    {k, 1, 12}], 1];
Table[vLen[allDiffIkos[[k]]], {k, 1, Length[allDiffIkos]}] //
 Union
```

Out[212]=
```
{0., 1.44721, 1.44721, 1.44721, 1.44721, 1.44721, 1.44721,
 1.44721, 1.44721, 2.34164, 2.34164, 2.34164, 2.34164, 2.34164,
 2.34164, 2.34164, 2.34164, 2.75276, 2.75276, 2.75276, 2.75276}
```

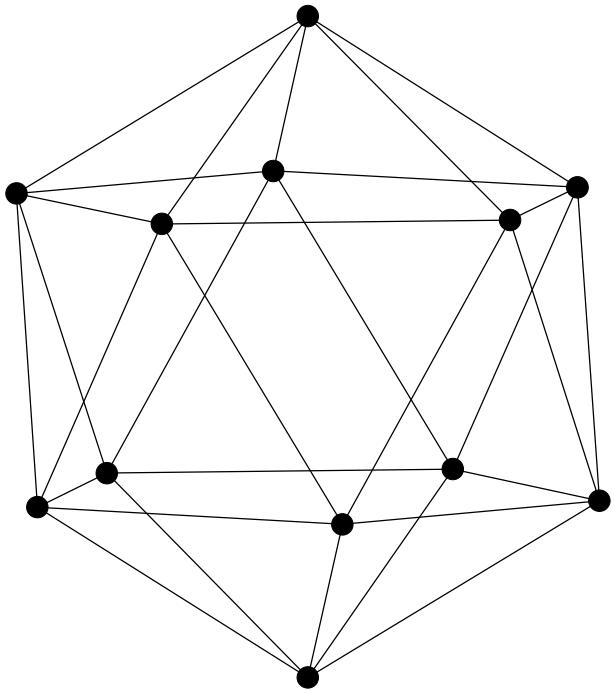
In[213]:=
```
sIkos = 1.4472135923940044
```

Out[213]=
```
1.44721
```

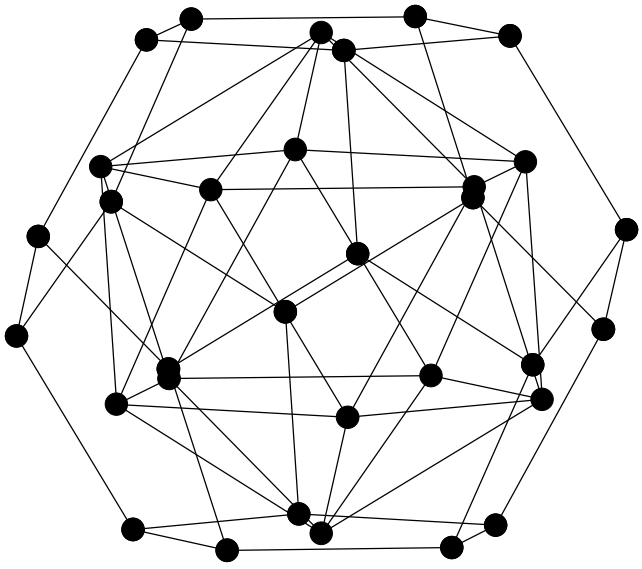
In[214]:=
```
seitenIkos =
 Map[Line,
  Drop[
   Union[
    Flatten[
     Table[
      Table[If[vLen[pDodNum[[j]] - pDodNum[[k]]] > 1.4 &&
         vLen[pDodNum[[j]] - pDodNum[[k]]] < 1.5,
        {pDodNum[[j]], pDodNum[[k]]}, hallo], {j, 1, k}], {k, 1, 12}],
      1]], 1]]
```

Out[214]=
```
{Line[{{-1.17082, -0.380423, 0.615537},
   {-1.17082, 0.380423, -0.615537}}], Line[
  {{-1.17082, -0.380423, 0.615537}, {-0.723607, -0.995959, -0.615537}}],
 Line[{{-1.17082, -0.380423, 0.615537}, {0., -1.23107, 0.615537}}],
 Line[{{-1.17082, 0.380423, -0.615537},
   {-0.723607, -0.995959, -0.615537}}],
 Line[{{-1.17082, 0.380423, -0.615537}, {0., 1.23107, -0.615537}}],
 Line[{{-0.723607, -0.995959, -0.615537},
   {0.723607, -0.995959, -0.615537}}], Line[
  {{-0.723607, 0.995959, 0.615537}, {-1.17082, -0.380423, 0.615537}}],
 Line[{{-0.723607, 0.995959, 0.615537},
   {-1.17082, 0.380423, -0.615537}}],
 Line[{{-0.723607, 0.995959, 0.615537}, {0., 1.23107, -0.615537}}],
 Line[{{0., -1.23107, 0.615537}, {-0.723607, -0.995959, -0.615537}}],
 Line[{{0., -1.23107, 0.615537}, {0.723607, -0.995959, -0.615537}}],
 Line[{{0., 0., -1.37638}, {-1.17082, 0.380423, -0.615537}}],
 Line[{{0., 0., -1.37638}, {-0.723607, -0.995959, -0.615537}}],
 Line[{{0., 0., -1.37638}, {0., 1.23107, -0.615537}}],
 Line[{{0., 0., -1.37638}, {0.723607, -0.995959, -0.615537}}],
 Line[{{0., 0., -1.37638}, {1.17082, 0.380423, -0.615537}}],
 Line[{{0., 0., 1.37638}, {-1.17082, -0.380423, 0.615537}}],
 Line[{{0., 0., 1.37638}, {-0.723607, 0.995959, 0.615537}}],
 Line[{{0., 0., 1.37638}, {0., -1.23107, 0.615537}}],
 Line[{{0., 0., 1.37638}, {0.723607, 0.995959, 0.615537}}],
 Line[{{0., 0., 1.37638}, {1.17082, -0.380423, 0.615537}}],
 Line[{{0.723607, -0.995959, -0.615537},
   {1.17082, 0.380423, -0.615537}}], Line[
  {{0.723607, 0.995959, 0.615537}, {-0.723607, 0.995959, 0.615537}}],
 Line[{{0.723607, 0.995959, 0.615537}, {0., 1.23107, -0.615537}}],
 Line[{{0.723607, 0.995959, 0.615537}, {1.17082, 0.380423, -0.615537}}],
 Line[{{1.17082, -0.380423, 0.615537}, {0., -1.23107, 0.615537}}], Line[
  {{1.17082, -0.380423, 0.615537}, {0.723607, -0.995959, -0.615537}}],
 Line[{{1.17082, -0.380423, 0.615537}, {0.723607, 0.995959, 0.615537}}],
 Line[{{1.17082, -0.380423, 0.615537}, {1.17082, 0.380423, -0.615537}}],
 Line[{{1.17082, 0.380423, -0.615537}, {0., 1.23107, -0.615537}}]}
```

In[215]:=
```
pointIL = Show[Graphics3D[{pointsI, seitenIkos}, Axes → False,
    AxesEdge → None, Boxed → False], ViewPoint → {100, 20, 10}];
```



In[216]:=
```
pointILDL =
  Show[Graphics3D[{pointsI, seitenIkos, pointD, lDodAll},
    Axes → False, AxesEdge → None, Boxed → False],
   ViewPoint → {100, 20, 10}];
```



## Andere Programmiertechnik:
## Dodekaeder aus dem Würfel durch Aufsetzen des goldenen Daches

In[217]:=
```
(* Remove["Global`*"] *)
```

In[218]:=
```
rho = (Sqrt[5] + 1) / 2;
```

In[219]:=
```
sW = 2; sD = sW / rho;
```

In[220]:=
```
p[1] := {1, 1, 1}; p[2] := {-1, 1, 1}; p[3] := {-1, -1, 1};
p[4] := {1, -1, 1};
```

In[221]:=
```
a = (sW - sD) / 2; b = 1; c = Sqrt[a^2 + b^2]; h = Sqrt[sD^2 - c^2];
h // N;
```

In[222]:=
```
p[5] = {0, sD / 2, 1 + h}; p[6] = {0, -sD / 2, 1 + h};
p[7] = {0, sD / 2, 1};
```

In[223]:=
```
A[fi1_] := {{1, 0, 0}, {0, Cos[fi1], -Sin[fi1]}, {0, Sin[fi1], Cos[fi1]}};
B[fi2_] := {{Cos[fi2], 0, -Sin[fi2]}, {0, 1, 0}, {Sin[fi2], 0, Cos[fi2]}};
```

In[224]:=
```
A[fi] // MatrixForm;
```

In[225]:=
```
B[f] // MatrixForm;
```

In[226]:=
```
fi1 = Pi / 2; fi2 = fi1;
f[k_, 1] := p[k];
f[k_, 2] := B[fi1].(A[fi1].p[k]);
f[k_, 3] := A[fi1].(B[fi1].p[k]);
f[k_, 4] := -f[k, 1];
f[k_, 5] := -f[k, 2];
f[k_, 6] := -f[k, 3];
```

In[233]:=
```
f[3, 1];
```

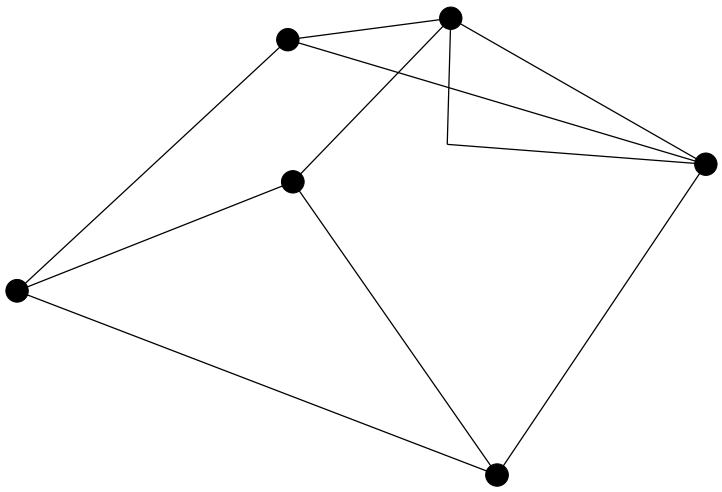
In[234]:=
```
point[t_] := Table[f[k, t], {k, 1, 6}]; point[1] // N;
```

In[235]:=
```
pointsP[t_] := Prepend[Map[Point, point[t]], PointSize[0.03]];
pointsP[1] // N;
```

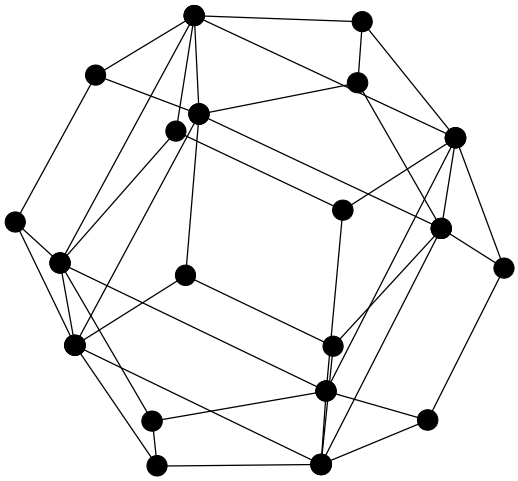
In[236]:=
```
linesP1[t_] :=
 Map[Line, {{f[1, t], f[2, t], f[3, t], f[4, t], f[1, t]},
   {f[1, t], f[5, t], f[6, t], f[4, t]},
   {f[2, t], f[5, t], f[6, t], f[3, t]}}];
linesP[1] // N;
```

In[237]:=
```
linesP[t_] :=
 Map[Line, {{f[2, t], f[3, t]}, {f[4, t], f[1, t]},
   {f[1, t], f[5, t], f[6, t], f[4, t]},
   {f[2, t], f[5, t], f[6, t], f[3, t]}}];
linesP[1] // N;
```

In[238]:=
```
Show[Graphics3D[{pointsP[1], linesP1[1],
    Line[{f[1, 1], f[7, 1], f[5, 1]}]}, Axes → None, Boxed → False]];
```



In[239]:=
```
sh[k_] := Show[Graphics3D[{pointsP[k], linesP1[k]}, Axes → None,
    Boxed → False], DisplayFunction → Identity];
Show[Evaluate[Flatten[Table[sh[k], {k, 1, 6}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {2, 1, 7}];
```



## Würfelfünfling ohne Durchdringung

In[241]:=
```
(* Remove["Global`*"] *)
```

In[242]:=
```
sW = 2;
```

In[243]:=
```
volW = sW^3;
diagFW = Sqrt[2] sW;
diagVW = Sqrt[3] sW;
rUW = diagVW / 2;
rIW = sW / 2;
rKW = diagVW / 2;
obW = 6 sW^2;
```

In[249]:=
```
sD = sW (Sqrt[5] - 1) / 2;
(*Dodekaederseite, Dodekaedersehne = Würfelseite sW   *)
Print["sW = ", sW // N];

s5Dod = sD; (*Dodekaederseite, Fünfeckseite*)
Print["s5Dod=sD = ", sD // N];

diag2Dod = s5Dod (1 + Sqrt[5]) / 2;  (*Dodekaedersehne = Würfelseite sW*)
Print["diag2Dod = ", diag2Dod // N];

s6Dod = s5Dod / Sqrt[1 + ((Sqrt[5] - 1) / 2)^2];
(*Sechseckseite, Fünfeckseite*)
Print["s6Dod = ", s6Dod // N];

rFDod = s6Dod; (*Dodekaederumkreisradius, Sechseckseite   *)
Print["rFDod = ", rFDod // N];

Hs6Dod = Sqrt[s6Dod^2 - (s5Dod / 2)^2];
(*Höhe Fünfeckseite bis Mittelpunkt   *)
Print["Hs6Dod = ", Hs6Dod // N];

Hs2Dod = Sqrt[s5Dod ^2 - (diag2Dod / 2)^2];
(*Höhe Dodekaedersehne bis Ecke*)
Print["Hs2Dod = ", Hs2Dod // N];

dKipp = s6Dod - Hs2Dod;
(*Kippdistanz zwischen Würfelkante und Fünfeckmitte Dodekaeder*)
Print["dKipp = ", dKipp // N];

φ1 = ArcSin[dKipp / (diagFW / 2)] // Simplify // N;
φ = ArcSin[dKipp / (diagFW / 2)];
(*Kippwinkel zwischen Würfel- und Dodekaederachse*)
Print["φ Winkeldifferenz = ", (φ1 / (2 Pi) 360.) // N, " Grad"];

φ2 = -Pi / 4 + φ;
Print["φ2 Kipp = ", (φ2 / (2 Pi) 360.) // N, " Grad"];

(*Halbe Flächendiagonale Würfel*)
Print["Halbe Flächendiagonale Würfel = ", diagFW / 2 // N];
```

```
sW = 2.

s5Dod=sD = 1.23607

diag2Dod = 2.

s6Dod = 1.05146

rFDod = 1.05146

Hs6Dod = 0.850651
```

Hs2Dod = 0.726543

dKipp = 0.32492

$\varphi$ Winkeldifferenz = 13.2825 Grad

$\varphi$2 Kipp = −31.7175 Grad

Halbe Flächendiagonale Würfel = 1.41421

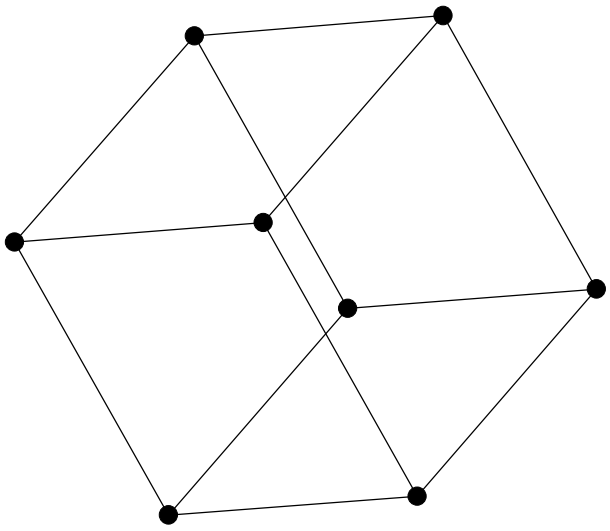
In[265]:=
```
p[1] := {1, 1, 1}; p[2] := {-1, 1, 1}; p[3] := {-1, -1, 1};
p[4] := {1, -1, 1};
p[5] := {1, 1, -1}; p[6] := {-1, 1, -1}; p[7] := {-1, -1, -1};
p[8] := {1, -1, -1};
```

In[267]:=
```
pKipp[k_, α_] := {{Cos[α], -Sin[α], 0}, {Sin[α], Cos[α], 0}, {0, 0, 1}}.
  ({{1, 0, 0}, {0, Cos[-φ2], -Sin[-φ2]}, {0, Sin[-φ2], Cos[-φ2]}}.p[k]);
pKipp[1, 0] // N;
```

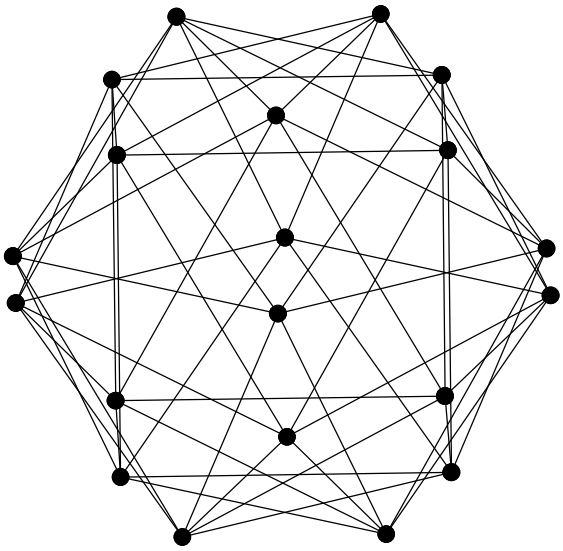
In[268]:=
```
pointsPNew[α_] := Prepend[Map[Point, Table[pKipp[k, α], {k, 1, 8}]],
  PointSize[0.03]];
pointsPNew[0] // N;
```

In[269]:=
```
linesPNew[α_] :=
 Map[Line,
  {{pKipp[1, α], pKipp[2, α], pKipp[3, α], pKipp[4, α], pKipp[1, α]},
   {pKipp[5, α], pKipp[6, α], pKipp[7, α], pKipp[8, α], pKipp[5, α]},
   {pKipp[1, α], pKipp[5, α]}, {pKipp[2, α], pKipp[6, α]},
   {pKipp[3, α], pKipp[7, α]}, {pKipp[4, α], pKipp[8, α]}}];
linesPNew[0] // N;
```

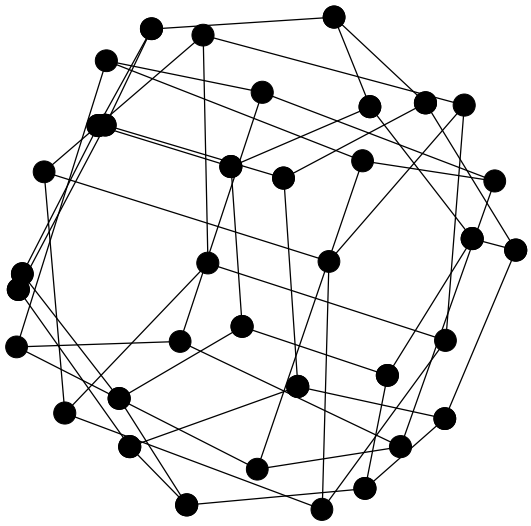
In[270]:=
```
Show[Graphics3D[{pointsPNew[1], linesPNew[1]}, Axes → None,
   Boxed → False], ViewPoint → {2, 100, 100}];
```



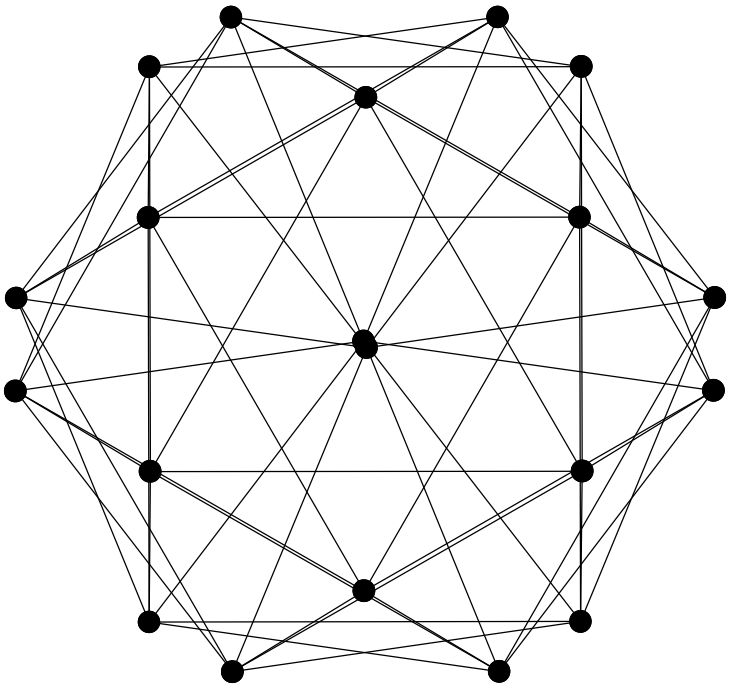
In[271]:=
```
sh[k_] := Show[Graphics3D[{pointsPNew[k], linesPNew[k]},
    Axes → None, Boxed → False], DisplayFunction → Identity];
Show[Evaluate[Flatten[Table[sh[k 2 Pi / 5], {k, 1, 5}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {2, 100, 100}];
```



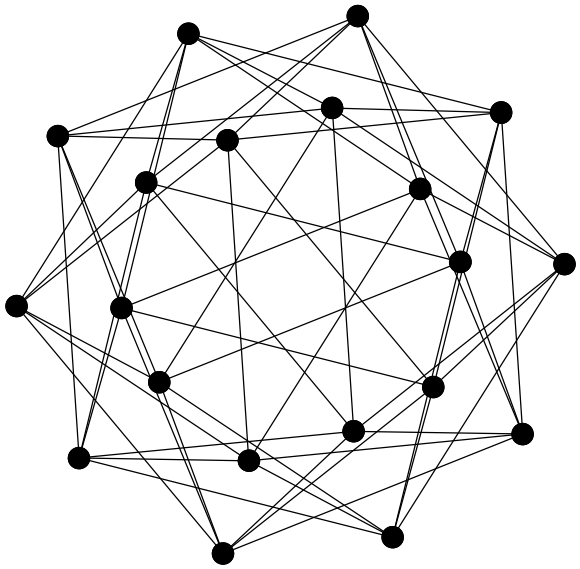
```
Show[nimm1, nimm2, sh[3]];
```



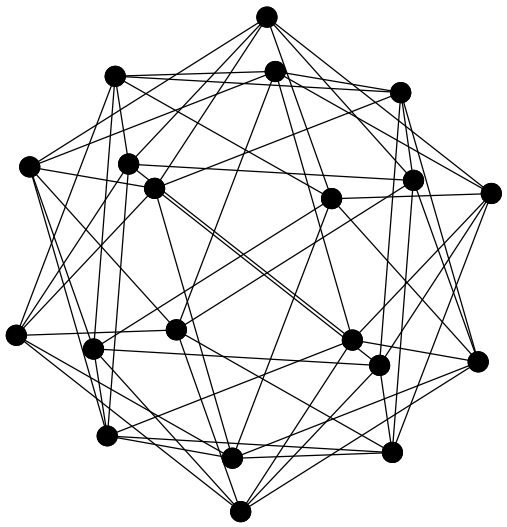
In[274]:=
```
Show[Evaluate[Flatten[Table[sh[k 2 Pi / 5], {k, 1, 5}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {2, 500, 100}];
```



In[275]:=
```
Show[Evaluate[Flatten[Table[sh[k 2 Pi / 5], {k, 1, 5}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {2, 5, 100}];
```



In[276]:=
```
Show[Evaluate[Flatten[Table[sh[k 2 Pi / 5], {k, 1, 5}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {60, 100, 100}];
```

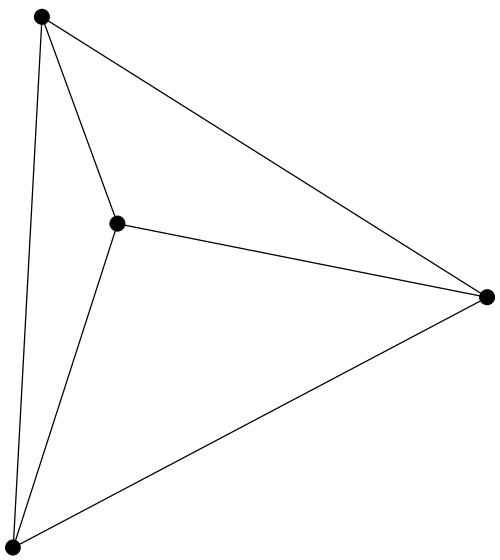


## Tetraederfünfling ohne Durchdringung

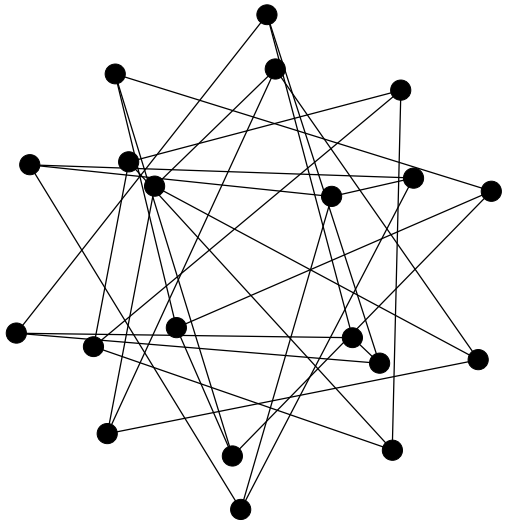
In[277]:=
```
pointsPNew1[α_] :=
 Prepend[Map[Point, {pKipp[1, α], pKipp[3, α], pKipp[6, α], pKipp[8, α]}],
  PointSize[0.03]];
pointsPNew1[0] // N;
```

In[278]:=
```
linesPNew1[α_] :=
 Map[Line,
  {{pKipp[1, α], pKipp[3, α], pKipp[6, α], pKipp[8, α], pKipp[1, α]},
   {pKipp[1, α], pKipp[6, α]}, {pKipp[3, α], pKipp[8, α]}}];
linesPNew1[0] // N;
```

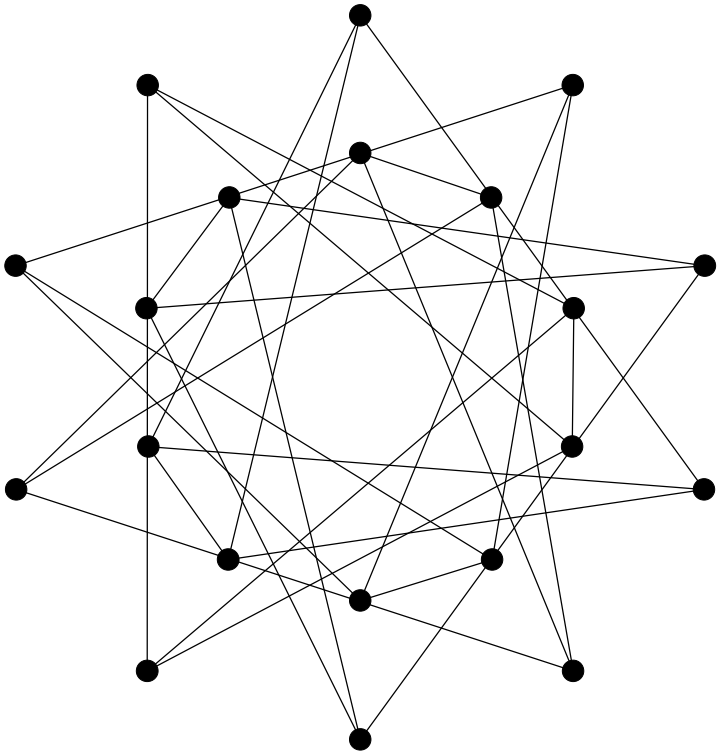
In[279]:=
```
Show[Graphics3D[{pointsPNew1[1], linesPNew1[1]}, Axes → None,
   Boxed → False], ViewPoint → {2, 100, 100}];
```



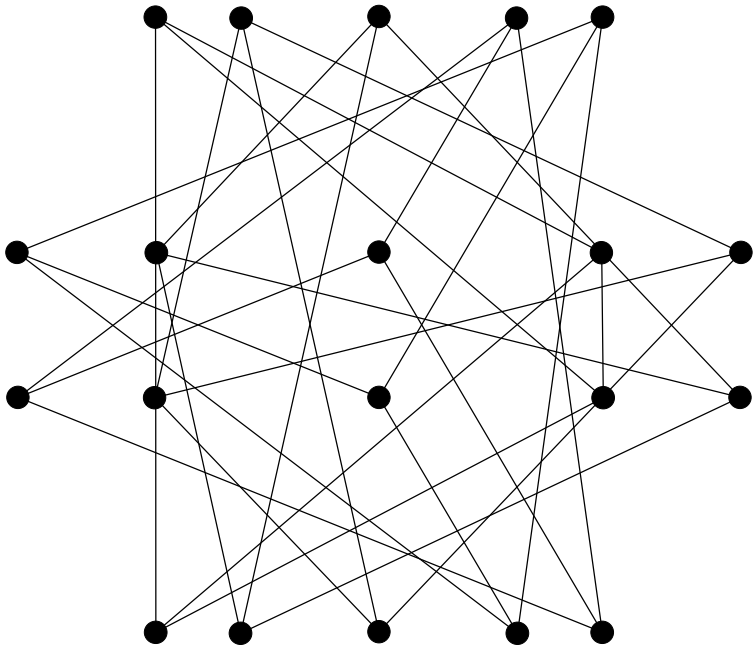
In[280]:=
```
sh[k_] := Show[Graphics3D[{pointsPNew1[k], linesPNew1[k]},
    Axes → None, Boxed → False], DisplayFunction → Identity];
Show[Evaluate[Flatten[Table[sh[k 2 Pi / 5], {k, 1, 5}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {60, 100, 100}];
```



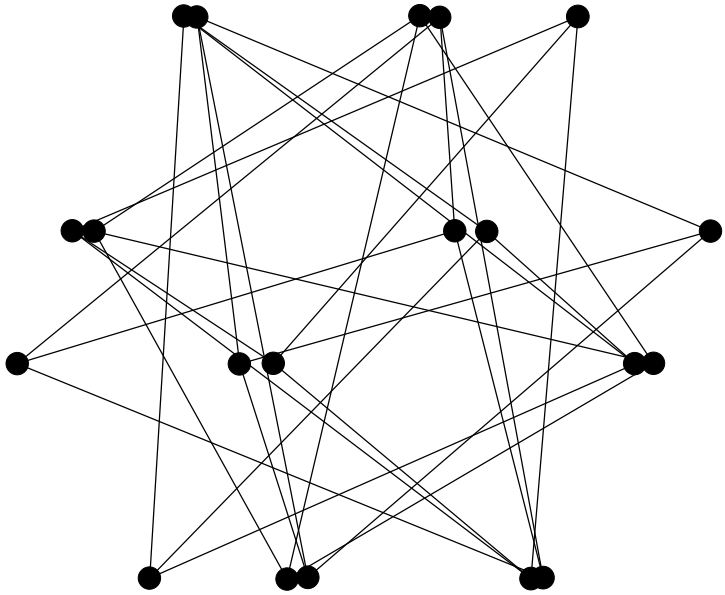
In[282]:=
```
Show[Evaluate[Flatten[Table[sh[k 2 Pi / 5], {k, 1, 5}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {0, 0, 100}];
```



In[283]:=
```
Show[Evaluate[Flatten[Table[sh[k 2 Pi / 5], {k, 1, 5}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {0, 100, 0}];
```



In[284]:=
```
Show[Evaluate[Flatten[Table[sh[k 2 Pi / 5], {k, 1, 5}]]],
  DisplayFunction → $DisplayFunction, ViewPoint → {100, 5, 0}];
```



## Weitere gegenseitige Beziehungen und Durchdringungen

### Selbst probieren

## Platonische Körper

Alle Ecken, Kanten und Flächen gleich

## Archimedische Körper

Alle Ecken gleich, 2 oder 3 verschiedene regelmässige n-Eck-Flächen

**Gewinnung aus den platonischen Körpern durch Abschleifen der Ecken**

**Gewinnung aus den platonischen Körpern durch Abschleifen der Kanten**

**Gewinnung aus den platonischen Körpern mittels Kantenmittlepunkten**

**Gewinnung aus schon vorhandenen archimedischen Körpern**

**Selbst probieren**

## Johnsonkörper

**Selbst probieren**

## Prismen und Antiprismen

**Selbst probieren**