Polyeder

K. Gerber

Körper, die nur von ebenen Flächen begrenzt erden, werden ***Vielflächner*** oder ***Polyeder*** genannt. Diejenigen Poplyeder, die keine nach innen einspringende Ecken haben, heißen ***konvexe Polyeder***.

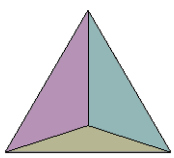
Für alle konvexen Polyeder gilt die ***Eulersche Polyederformel***: **E - K +F =2** (E: Anzahl der Ecken; K: Anzahl der Kanten; F: Anzahl der Flächen)

Ein Polyeder, dass von gleichseitigen kongruenten Flächen begrenzt wird, heißt ***reguläres Polyeder*** oder ***Platonischer Körper***. Davon gibt es fünf verschiedene.

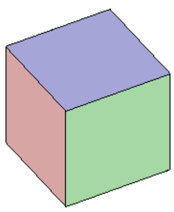
Bilder

Es gibt fünf Platonische Körper

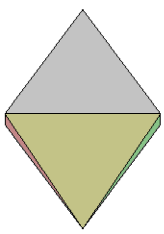
Tetraeder



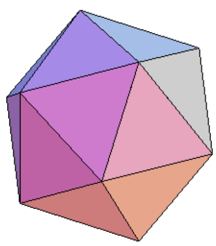
Hexaeder



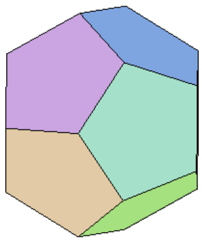
Oktaeder



Ikosaeder

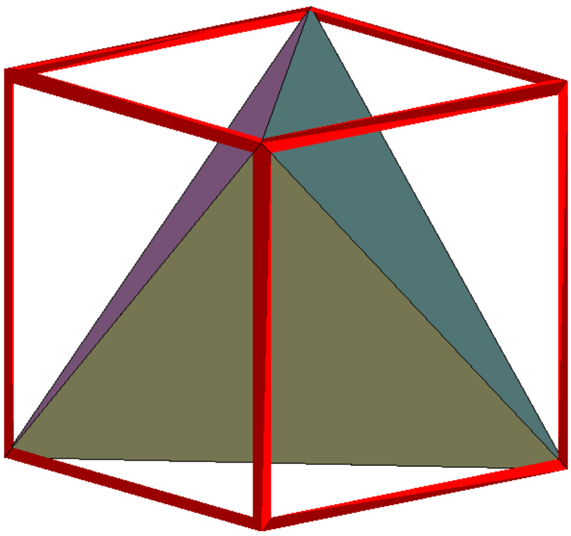


Dodekaeder

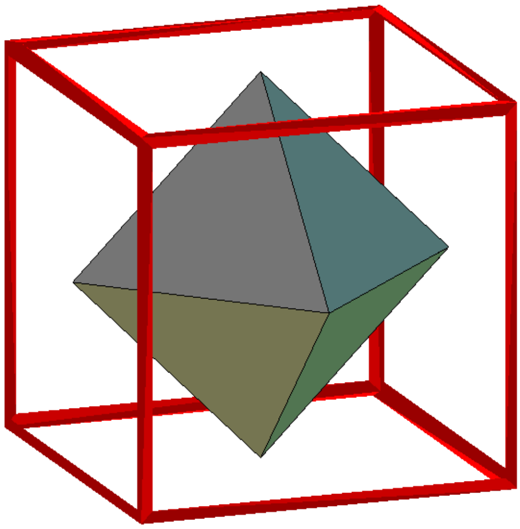


Sie können dem Würfel ein- oder umschrieben werden:

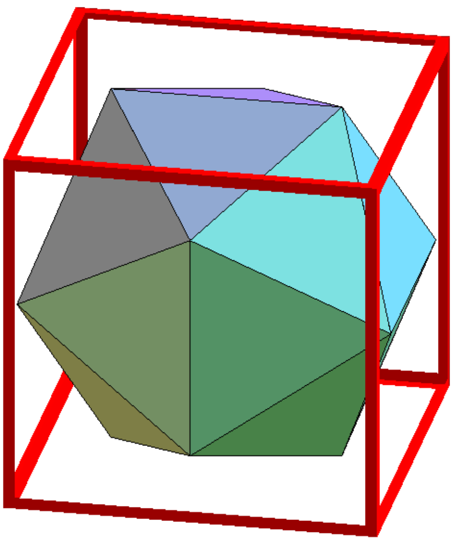
Tetraeder



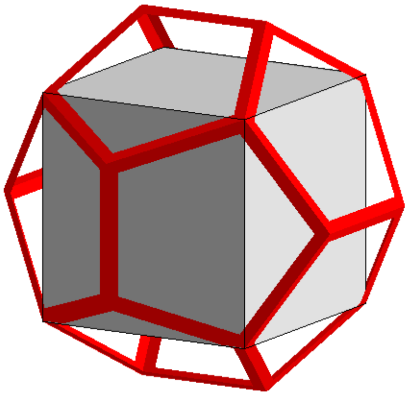
Oktaeder



Ikosaeder

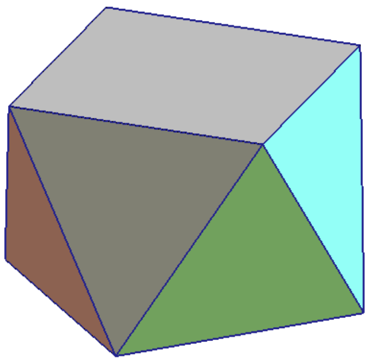


Dodekaeder

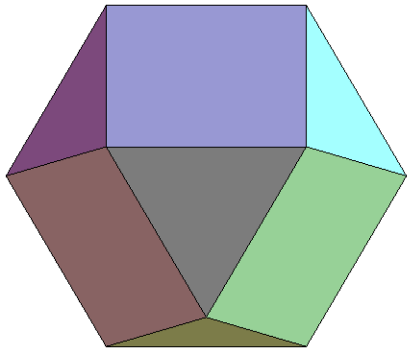


Weitere Polyeder:

Antiprisma



Kuboktaeder



Zum Nachweis der Eulerschen Polyederformel:

	E	K	F	E-K+F
Tetraeder	4	6	4	**2**
Hexaeder	8	12	6	**2**
Oktaeder	6	12	8	**2**
Ikosaeder	12	30	20	**2**
Dodekaeder	20	30	12	**2**
Kuboktaeder	12	24	14	**2**
Antiprisma	8	16	10	**2**

Hexaeder und Oktaeder sowie Ikosaeder und Dodekaeder sind zueinander duale Körper.

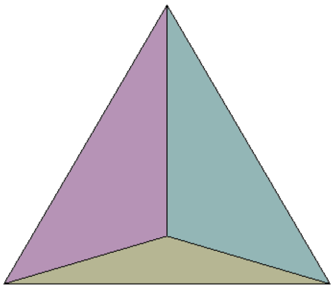
Zeichnen der Platonischen Körper

```
> restart;
  with(geom3d):
```

Flächen:

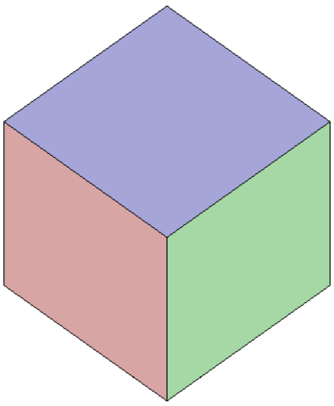
```
> draw(tetrahedron(t),title=Tetraeder);
```

Tetraeder

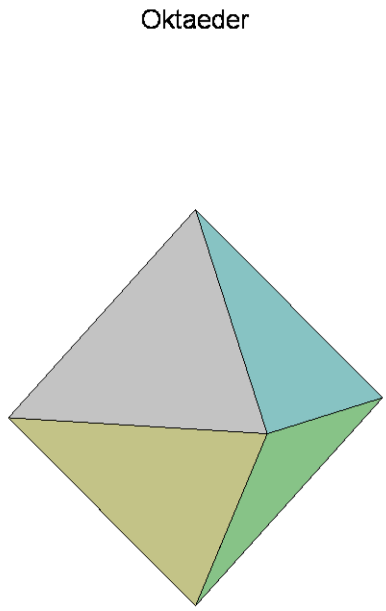


```
> draw(hexahedron(t),title=Hexaeder);
```

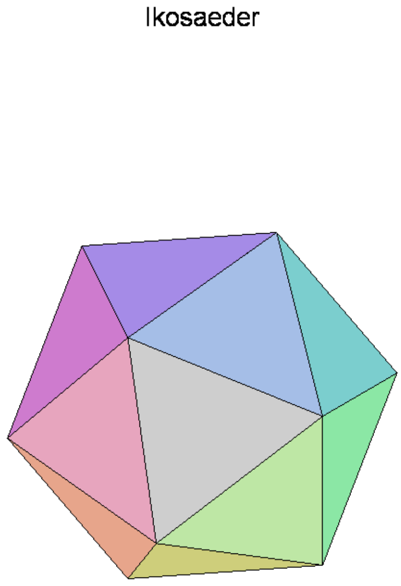
Hexaeder



```
> draw(octahedron(t),title=Oktaeder);
```

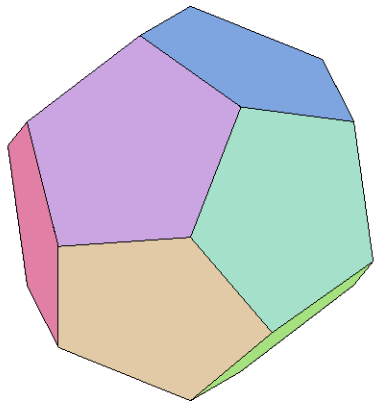



```
> draw(icosahedron(t),title=Ikosaeder);
```




```
> draw(dodecahedron(t),title=Dodekaeder);
```

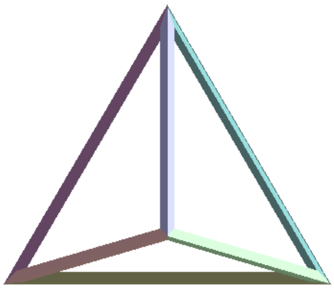
Dodekaeder



Kantenmodelle:

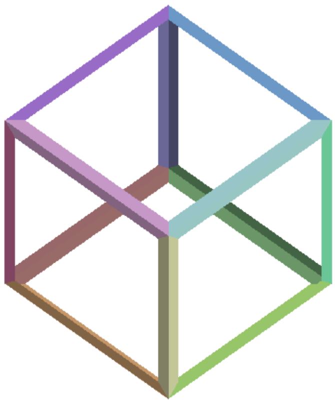
```
> draw(tetrahedron(t),title=Tetraeder,cutout=7/8,lightmodel=light4);
```

Tetraeder



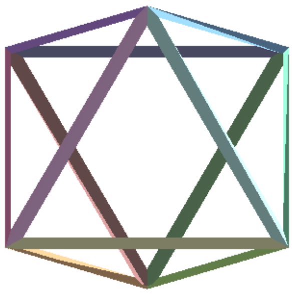
```
> draw(hexahedron(t),title=Hexaeder,cutout=7/8,lightmodel=light4);
```

Hexaeder

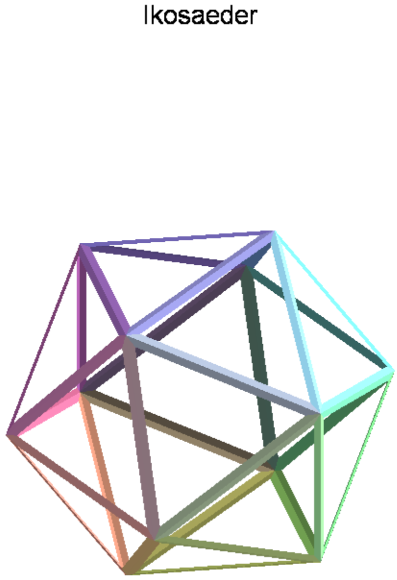


```
> draw(octahedron(t),title=Oktaeder,cutout=7/8,lightmodel=light4);
```

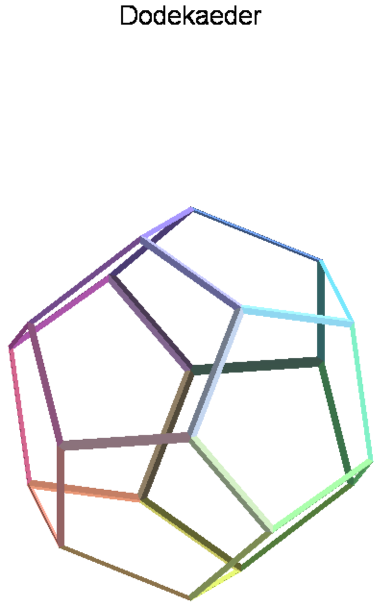
Oktaeder



```
> draw(icosahedron(t),title=Ikosaeder,cutout=7/8,lightmodel=light4);
```



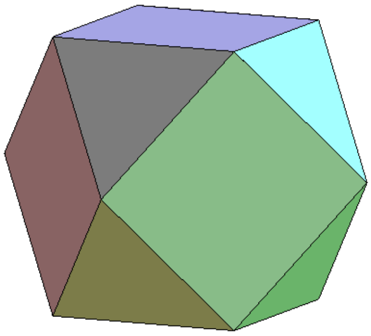

```
> draw(dodecahedron(t),title=Dodekaeder,cutout=7.5/8,lightmodel
=light4);
```



Weitere konvexe Polyeder:

```
> draw(cuboctahedron(t),title=Kuboktaeder,lightmodel=light4);
```

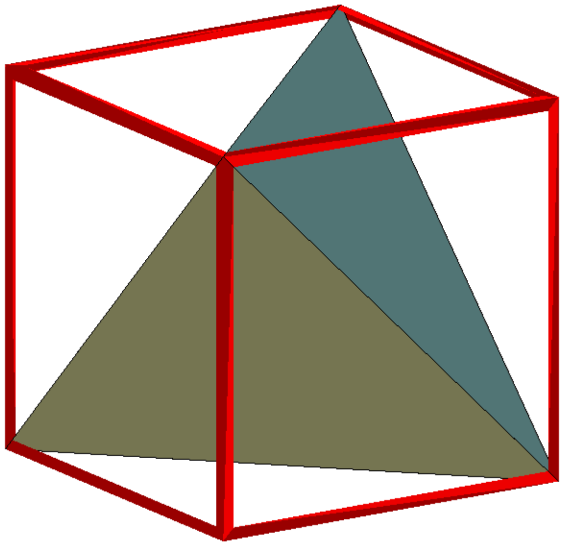
Kuboktaeder



Ein- und Umschreibungen:

```
> tetrahedron(tetra,point(o,0,0,0),side=sqrt(8)):
  hexahedron(hexa,point(o,0,0,0),side=2):
  draw({tetra,hexa(cutout=7.5/8,color=red)},lightmodel=light4,v
  iew=[-1..1,-1..1,-1..1],title=Tetraeder);
```

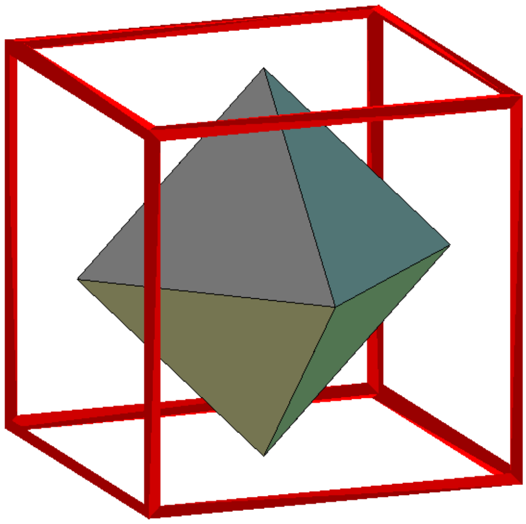
Tetraeder



```
> octahedron(octa,point(o,0,0,0),side=sqrt(2)):hexahedron(hexa,
```

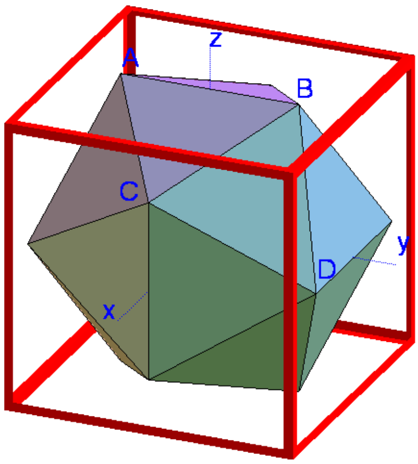
```
point(o,0,0,0),side=2):
draw({octa,hexa(cutout=7.5/8,color=red)},lightmodel=light4,view=[-1..1,-1..1,-1..1],title=Oktaeder);
```

Oktaeder



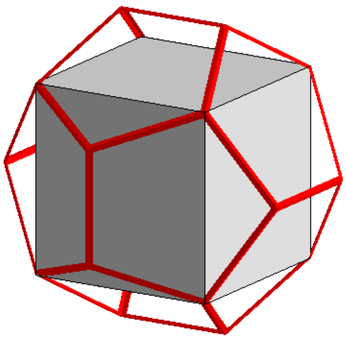
```
> icosahedron(ikosa,point(o,0,0,0),side=sqrt(5)-1):
hexahedron(hexa,point(o,0,0,0),side=2):
P1:=draw({ikosa,hexa(cutout=7.5/8,color=red)},lightmodel=light4,view=[-1.3..1.05,-1.05..1.5,-1.05..1.3],title=Ikosaeder):
P2 := plottools[line]([0,0,0], [-1.3,0,0], color=blue, linestyle=2):
P3 := plottools[line]([0,0,0], [0,1.5,0], color=blue, linestyle=2):
P4 := plottools[line]([0,0,0], [0,0,1.3], color=blue, linestyle=2):
P5 :=
plots[textplot3d]({[-1.3,0,0.1,`y`],[0.1,1.5,0,`x`],[0,0,1.3,`z`],[0.6,0,1.05,`A`],[-0.6,0,1.05,`B`],[0.2,1,0.6,`C`],[-1,0.6,0.1,`D`]},color=blue):
plots[display]({P1,P2,P3,P4,P5});
```

Ikosaeder




```
> dodecahedron(dode,point(o,0,0,0),side=sqrt(5)-1):
  hexahedron(hexa,point(o,0,0,0),side=2):
  draw({dode(cutout=7.5/8,color=red),hexa(color=grey)},lightmod
  el=light4,view=[-2..2,-2..2,-2..2],title=Dodekaeder);
```

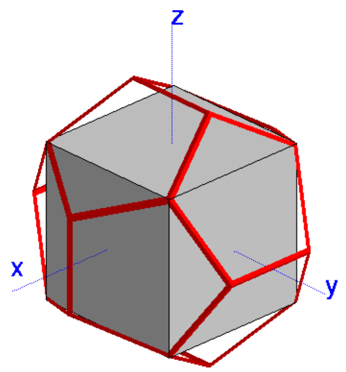
Dodekaeder



```
> dodecahedron(dode,point(o,0,0,0),side=sqrt(5)-1):
  hexahedron(hexa,point(o,0,0,0),side=2):
  P1:=draw({dode(cutout=7.5/8,color=red),hexa(color=grey)},ligh
```

```
tmodel=light4,view=[-2..2.5,-2..2.5,-2..2.5],title=Dodekaeder
):
P2 := plottools[line]([0,0,0], [2.5,0,0], color=blue,
linestyle=2):
P3 := plottools[line]([0,0,0], [0,2.5,0], color=blue,
linestyle=2):
P4 := plottools[line]([0,0,0], [0,0,2.5], color=blue,
linestyle=2):
P5 :=
plots[textplot3d]({[2.5,0,0.2,`x`],[0,2.5,0,`y`],[0,0,2.5,`z`
]},color=blue):
plots[display]({P1,P2,P3,P4,P5});
```

Dodekaeder



Das dem Würfel eingeschreibene Tetraeder

```
> restart:with(linalg):with(student):with(plots):with(geom3d):
Warning, new definition for norm
Warning, new definition for trace
Warning, new definition for distance
Warning, new definition for inverse
Warning, new definition for midpoint
```

Die Koordinaten:

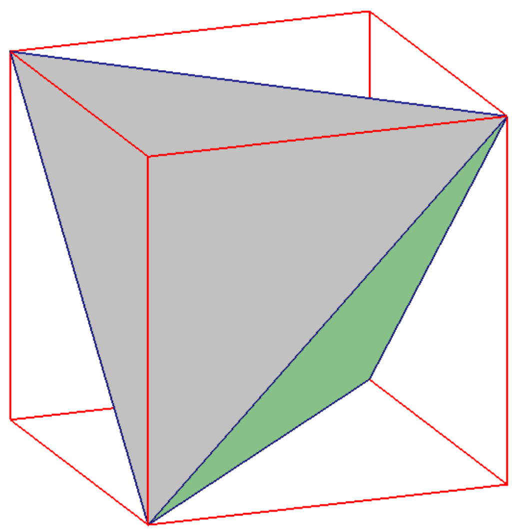
```
> A1:=[a,-a,-a]: A2:=[a,a,-a]: A3:=[-a,a,-a]: A4:=[-a,-a,-a]:
  B1:=[a,-a,a]: B2:=[a,a,a]: B3:=[-a,a,a]: B4:=[-a,-a,a]:
```

Die kantenlänge des Würfels ist 2a:

```
> a:=3:
```

Zeichnung des Tetraeders:

```
> T1:=plottools[polygon]([A4,A2,B1],style=patch,linestyle=7):
  T2:=plottools[polygon]([A4,A2,B3],style=patch,linestyle=7):
  T3:=plottools[polygon]([B1,B3,A2],style=patch,linestyle=7):
  T4:=plottools[polygon]([B1,B3,A4],style=patch,linestyle=7):
  K:=pointplot3d([A4,A2,B1,A4,B3,A2,B3,B1],style=line,color=nav
  y,thickness=2):
  W:=pointplot3d([A1,A2,A3,A4,A1,B1,B2,A2,B2,B3,A3,B3,B4,A4,B4,
  B1],style=line,color=red,thickness=2):
  display({T1,T2,T3,T4,K,W},scaling=constrained);
```



Winkel zwischen den Kanten (im gleichseitigen Dreieck ist dies klar):

```
> u:=A2-B1;
  v:=B3-B1;
  w:=A4-B1;
  alpha =
  arccos(abs(multiply(u,v))/(norm(u,2)*norm(v,2)))*180/Pi;
```

$$u := [0, 6, -6]$$
$$v := [-6, 6, 0]$$
$$w := [-6, 0, -6]$$
$$\alpha = 60$$

Winkel zwischen zwei Ebenen:

```
> n1:=crossprod(A2-B1,A4-B1);
  n2:=crossprod(B3-B1,A4-B1);
  beta =
```

```
evalf(arccos(abs(multiply(n1,n2)/(norm(n1,2)*norm(n2,2))))*18
0/Pi);
```

$$n1 := [-36, 36, 36]$$
$$n2 := [-36, -36, 36]$$
$$\beta = 70.52877934$$

Flächeninhalt einer Seitenfläche:

```
> A := 1/2*norm(crossprod(u,v),2);
```

$$A := 18\sqrt{3}$$

Höhe des Tetraeders:

```
> n3:=crossprod(u,w);
  t:=multiply(n3,[x,y,z]);
  c:=subs(x=a,y=a,z=-a,t);
  HNF:=simplify((t-c)/norm(n2,2))=0;
```

$$n3 := [-36, 36, 36]$$
$$t := -36\,x + 36\,y + 36\,z$$
$$c := -108$$
$$HNF := \frac{1}{3}(-x + y + z + 3)\sqrt{3} = 0$$

```
> h:=subs(x=-a,y=a,z=a,lhs(HNF));
```

$$h := 4\sqrt{3}$$

Volumen:

```
> V1:=1/6*abs(det(matrix([u,v,w])));
```

$$V1 := 72$$

```
> V2:=1/3*A*h;
```

$$V2 := 72$$

```
> S:=1/3*(B1+A2+A4);
  r_Innkugel:=norm(S,2);
  A_Koerper:=4*A;
  V3:=1/3*r_Innkugel*A_Koerper;
```

$$S := [1, -1, -1]$$
$$r\_Innkugel := \sqrt{3}$$
$$A\_Koerper := 72\sqrt{3}$$
$$V3 := 72$$

Welche ebene Fläche schneidet die x-y-Ebene aus dem Tetraeder aus?

```
> g1:=B1+s*u;
```

$$g1 := [3, -3, 3] + s\,[0, 6, -6]$$

```
> solve(1-2*s=0,s);
```

$$\frac{1}{2}$$

```
> C1:=subs(s=1/2,g1);
```

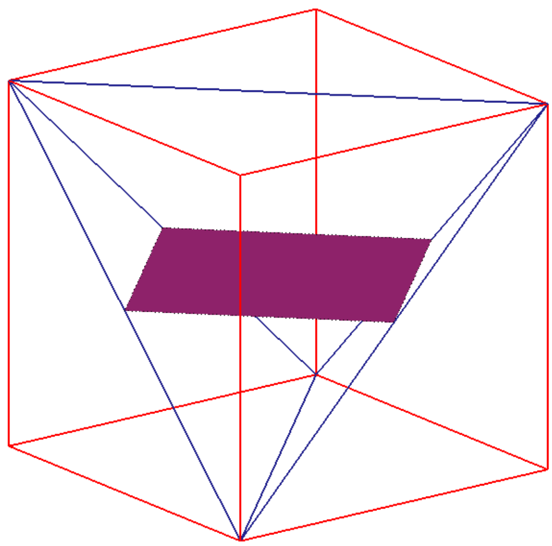
$$C1 := [3, 0, 0]$$

Es entsteht ein Quadrat mit den weiteren Eckpunkten:

```
> C2:=[0,a,0]: C3:=[-a,0,0]: C4:=[0,-a,0]:
```

Zum Zeichnen:

```
> Ebene:=plottools[polygon]([C1,C2,C3,C4],style=patch,linestyle
  =7,color=maroon):
  K:=pointplot3d([A4,A2,B1,A4,B3,A2,B3,B1],style=line,color=nav
  y,thickness=2):
  W:=pointplot3d([A1,A2,A3,A4,A1,B1,B2,A2,B2,B3,A3,B3,B4,A4,B4,
  B1],style=line,color=red,thickness=2):
  display({Ebene,K,W},scaling=constrained);
```



Maple kann es auch:

```
> tetrahedron(tetra,point(o,0,0,0),side=sqrt(2*(2*a)^2)):
  hexahedron(hexa,point(o,0,0,0),side=2*a):
  Volumen = volume(tetra);
  Seitenlänge = sides(tetra);
  Oberfläche = area(tetra);
  Radius = InRadius(tetra);
  draw({tetra,hexa(cutout=7.5/8,color=red)},lightmodel=light4,v
  iew=[-a..a,-a..a,-a..a],title=Tetraeder);
```

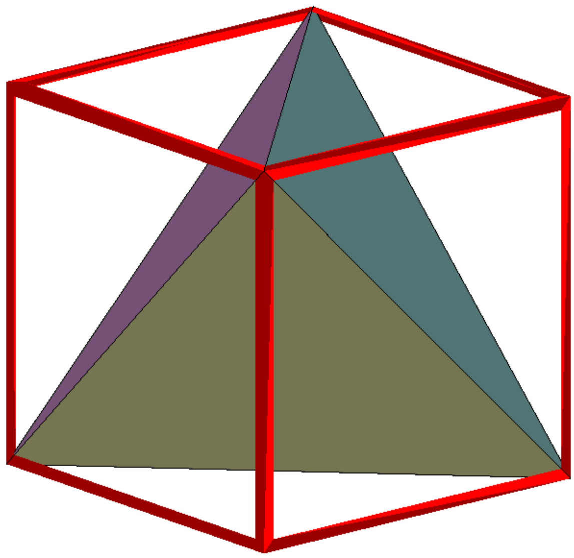
$$Volumen = 72$$

$$Seitenlänge = 6\sqrt{2}$$

$$Oberfläche = 72\sqrt{3}$$

$$Radius = \frac{1}{2}\sqrt{6}\,\sqrt{2}$$

Tetraeder



>

Das dem Würfel eingeschreibene Oktaeder

```
> restart:with(linalg):with(student):with(plots):with(geom3d):
Warning, new definition for norm
Warning, new definition for trace
Warning, new definition for distance
Warning, new definition for inverse
Warning, new definition for midpoint
```

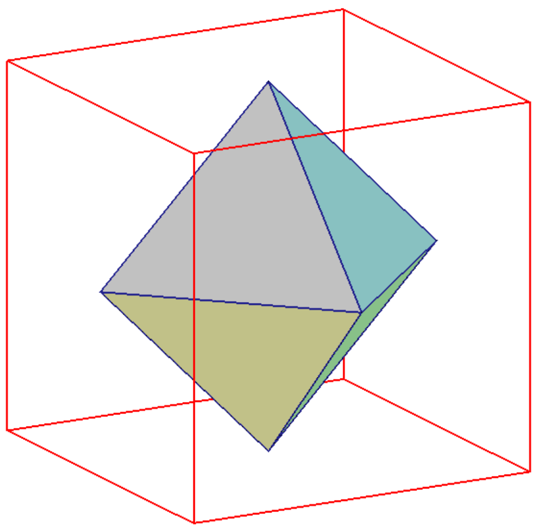
Die Koordinaten:

```
> A1:=[1,-1,-1]: A2:=[1,1,-1]: A3:=[-1,1,-1]: A4:=[-1,-1,-1]:
  B1:=[1,-1,1]: B2:=[1,1,1]: B3:=[-1,1,1]: B4:=[-1,-1,1]:
  C1:=[1,0,0]: C2:=[0,1,0]: C3:=[-1,0,0]: C4:=[0,-1,0]:
  C5:=[0,0,-1]: C6:=[0,0,1]:
```

Zeichnung des Oktaeders:

```
> T1:=plottools[polygon]([C1,C2,C6],style=patch,linestyle=7):
  T2:=plottools[polygon]([C2,C3,C6],style=patch,linestyle=7):
  T3:=plottools[polygon]([C3,C4,C6],style=patch,linestyle=7):
  T4:=plottools[polygon]([C4,C1,C6],style=patch,linestyle=7):
  T5:=plottools[polygon]([C1,C2,C5],style=patch,linestyle=7):
  T6:=plottools[polygon]([C2,C3,C5],style=patch,linestyle=7):
  T7:=plottools[polygon]([C3,C4,C5],style=patch,linestyle=7):
  T8:=plottools[polygon]([C4,C1,C5],style=patch,linestyle=7):
```

```
K:=pointplot3d([C5,C1,C6,C3,C5,C2,C6,C4,C5,C1,C2,C3,C4,C1],style=line,color=navy,thickness=2):
W:=pointplot3d([A1,A2,A3,A4,A1,B1,B2,A2,B2,B3,A3,B3,B4,A4,B4,B1],style=line,color=red,thickness=2):
display({T1,T2,T3,T4,T5,T6,T7,T8,K,W},scaling=constrained);
```



Winkel zwischen den Kanten (im gleichseitigen Dreieck ist dies klar):

```
> u:=C1-C6;
  v:=C2-C6;
  w:=C3-C6;
  alpha = arccos(abs(multiply(u,v))/(norm(u,2)*norm(v,2)))*180/Pi;
```

$$u := [1, 0, -1]$$

$$v := [0, 1, -1]$$

$$w := [-1, 0, -1]$$

$$\alpha = 60$$

Winkel zwischen zwei Ebenen:

```
> n1:=crossprod(u,v);
  n2:=crossprod(v,w);
  beta = evalf(arccos(abs(multiply(n1,n2)/(norm(n1,2)*norm(n2,2))))*180/Pi);
```

$$n1 := [1, 1, 1]$$

$$n2 := [-1, 1, 1]$$

$$\beta = 70.52877934$$

Flächeninhalt einer Seitenfläche:

```
> A := 1/2*norm(crossprod(u,v),2);
```

$$A := \frac{1}{2}\sqrt{3}$$

Höhe einer Pyramide:

```
> h:=1;
```

$$h := 1$$

Volumen:

```
> V1:=4*1/6*abs(det(matrix([C6-C1,C2-C1,C3-C1])));
```

$$V1 := \frac{4}{3}$$

```
> V2:=2*1/3*2*h;
```

$$V2 := \frac{4}{3}$$

```
> S:=1/3*(C1+C2+C6);
  r_Innkugel:=norm(S,2);
  A_Koerper:=8*A;
  V3:=1/3*r_Innkugel*A_Koerper;
```

$$S := \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$$

$$r\_Innkugel := \frac{1}{3}\sqrt{3}$$

$$A\_Koerper := 4\sqrt{3}$$

$$V3 := \frac{4}{3}$$

Maple kann es auch:

```
> octahedron(octa,point(o,0,0,0),side=sqrt(2)):hexahedron(hexa,
  point(o,0,0,0),side=2):
  Volumen = volume(octa);
  Seitenlänge = sides(octa);
  Oberfläche = area(octa);
  Radius = InRadius(octa);
  draw({octa,hexa(cutout=7.5/8,color=red)},lightmodel=light4,vi
  ew=[-1..1,-1..1,-1..1],title=Oktaeder);
```

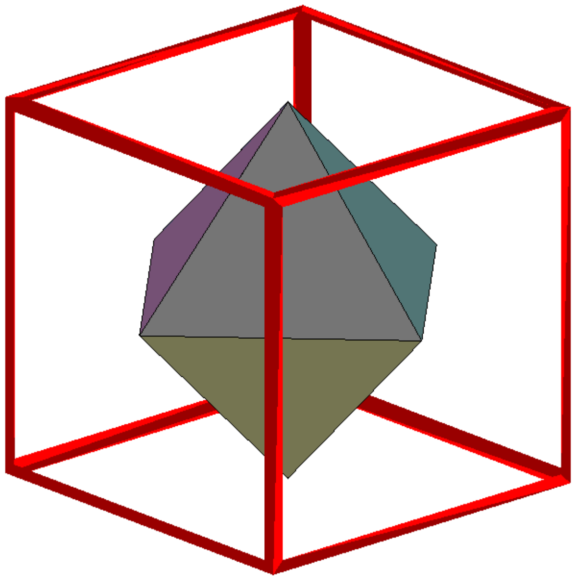
$$Volumen = \frac{4}{3}$$

$$Seitenlänge = \sqrt{2}$$

$$Oberfläche = 4\sqrt{3}$$

$$Radius = \frac{1}{3}\sqrt{3}$$

Oktaeder



Das dem Würfel eingeschreibene Ikosaeder

>

>

>

Das dem Würfel umschreibene Dodekaeder

>

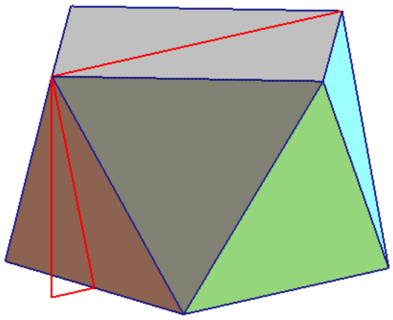
>

>

Das Antiprisma

```
> restart:with(linalg):with(student):with(plots):with(geom3d):
Warning, new definition for norm
Warning, new definition for trace
Warning, new definition for distance
Warning, new definition for inverse
Warning, new definition for midpoint
```

Antiprisma



Die Koordinaten der Eckpunkte des Bodens sind:

```
> A1:=[5,-5,0]: A2:=[5,5,0]: A3:=[-5,5,0]: A4:=[-5,-5,0]:
```

Nun fehlen noch die Koordinaten der Eckpunkte des Deckels.
Mit Pythagoras findet man zunächst die x-Koordinate des Punktes B1:

```
> l:=1/2*sqrt(10^2+10^2);
```

$$l := 5\sqrt{2}$$

Außerdem ist die Flächenhöhe im gleichseitigen Dreieck A1 A2 B1 mit Pythagoras zu berechnen:

```
> h_Dreieck:=sqrt(10^2-5^2);
```

$$h\_Dreieck := 5\sqrt{3}$$

Für die Höhe des Punktes B1 über der xy-Ebene wird Pythagoras zum dritten Mal bemüht:

```
> gl:=(h_Dreieck)^2=h^2+(l-5)^2;
```

$$gl := 75 = h^2 + (5\sqrt{2} - 5)^2$$

```
> gL:=solve(gl,h);
```

$$gL := 5\,2^{(3/4)}, -5\,2^{(3/4)}$$

```
> h:=gL[1];
```

$$h := 5\,2^{(3/4)}$$

Die Koordinaten sind also:

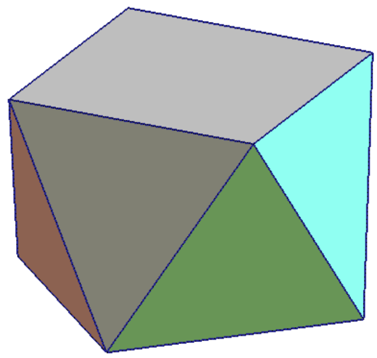
```
> l:=5*sqrt(2): h:=5*sqrt(2*sqrt(2)):
  B1:=[l,0,h]: B2:=[0,l,h]: B3:=[-l,0,h]: B4:=[0,-l,h]:
```

Zeichnung des Antiprismas:

```
> T1:=plottools[polygon]([A1,B1,A2],style=patch,linestyle=7):
```

```
T2:=plottools[polygon]([B1,A2,B2],style=patch,linestyle=7):
T3:=plottools[polygon]([A2,B2,A3],style=patch,linestyle=7):
T4:=plottools[polygon]([B2,A3,B3],style=patch,linestyle=7):
T5:=plottools[polygon]([A3,B3,A4],style=patch,linestyle=7):
T6:=plottools[polygon]([B3,A4,B4],style=patch,linestyle=7):
T7:=plottools[polygon]([A4,B4,A1],style=patch,linestyle=7):
T8:=plottools[polygon]([B4,A1,B1],style=patch,linestyle=7):
Boden:=plottools[polygon]([A1,A2,A3,A4],style=patch,linestyle
=7,color=grey):
Deckel:=plottools[polygon]([B1,B2,B3,B4],style=patch,linestyl
e=7,color=grey):
Kanten:=pointplot3d([A1,B1,A2,B2,A3,B3,A4,B4,A1,A2,A3,A4,A1,B
1,B2,B3,B4,B1],style=line,color=navy,thickness=2):
Pythagoras:=pointplot3d([B1,[5,0,0],[1,0,0],B1,[-1,0,h]],styl
e=line,color=red,thickness=2):
display({T1,T2,T3,T4,T5,T6,T7,T8,Deckel,Boden,Kanten},scaling
=constrained,title=Antiprisma,lightmodel=light4);
```

Antiprisma



Winkel zwischen den Kanten:

```
> u:=A1-B1:
v:=A2-B1:
w:=A2-B2:
alpha =
evalf(arccos(abs(multiply(u,v))/(norm(u,2)*norm(v,2)))*180/Pi
);
```

$$\alpha = 59.99999998$$

Winkel zwischen zwei Ebenen:

```
> n1:=crossprod(u,v);
  n2:=[0,0,1];
  beta =
  evalf(arccos(abs(multiply(n1,n2)/(norm(n1,2)*norm(n2,2))))*18
  0/Pi);
```

$$n1 := [50\,2^{(3/4)}, 0, -50\sqrt{2}+50]$$

$$n2 := [0, 0, 1]$$

$$\beta = 76.16383954$$

```
> n3:=crossprod(v,w);
  gamma =
  evalf(arccos(abs(multiply(n1,n3)/(norm(n1,2)*norm(n3,2))))*18
  0/Pi);
```

$$n3 := [-25\,2^{(3/4)} + 5\,2^{(3/4)}(-5\sqrt{2}+5), -25\,2^{(3/4)} + 5\,2^{(3/4)}(-5\sqrt{2}+5), (-5\sqrt{2}+5)^2 - 25]$$

$$\gamma = 52.44839710$$

Flächeninhalt einer Seitenfläche:

```
> A := simplify(1/2*norm(crossprod(u,v),2)); evalf(A);
```

$$A := 25\sqrt{3}$$

$$43.30127020$$

Das Antiprisma kann aus 10 Pyramiden zusammengesetzt werden. Den Flächeninhalt der Grundfläche der 8 gleichen Pyramiden haben wir mit A schon berechnet. Es fehlt noch der Abstand dieser Flächen vom Punkt [0,0,h/2]. Der Flächeninhalt der Grundflächen der beiden weiteren Pyramiden beträgt 20^2 FE; ihre Höhen sind h/2.

Abstand des Punktes [0,0,h/2] von einer Seitenfläche:

```
> S1:=1/3*(A1+A2+B1);
```

$$S1 := \left[\frac{5}{3}\sqrt{2} + \frac{10}{3}, 0, \frac{5}{3}2^{(3/4)}\right]$$

```
> d1:=simplify(norm(S1-[0,0,h/2],2));
  evalf(d1);
```

$$d1 := \frac{5}{6}\sqrt{24 + 18\sqrt{2}}$$

$$5.860404098$$

```
> V:='8*(1/3)*A*d1+2*(1/3)*10^2*(h/2)';
  'V'=evalf(V);
```

$$V := \frac{8}{3}A\,d1 + \frac{100}{3}h$$

$$V = 956.9999819$$

```
>
```

Das abgeschnittene Tetraeder

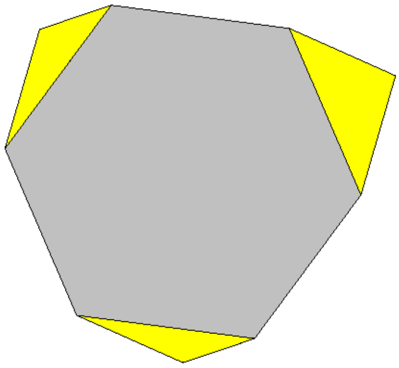
```
> restart:with(linalg):with(student):with(plots):with(geom3d):
Warning, new definition for norm
Warning, new definition for trace
Warning, new definition for distance
Warning, new definition for inverse
Warning, new definition for midpoint
```

Die Koordinaten:

```
> A1:=[3,-3,-3]: A2:=[3,3,-3]: A3:=[-3,3,-3]: A4:=[-3,-3,-3]:
  B1:=[3,-3,3]: B2:=[3,3,3]: B3:=[-3,3,3]: B4:=[-3,-3,3]:
> a1:=[1,1,-3]: a2:=[-1,-1,-3]:
  b1:=[3,1,-1]: b2:=[1,3,-1]: b3:=[-3,-1,-1]: b4:=[-1,-3,-1]:
  c1:=[1,-3,1]: c2:=[3,-1,1]: c3:=[-1,3,1]: c4:=[-3,1,1]:
  d1:=[1,-1,3]: d2:=[-1,1,3]:
```

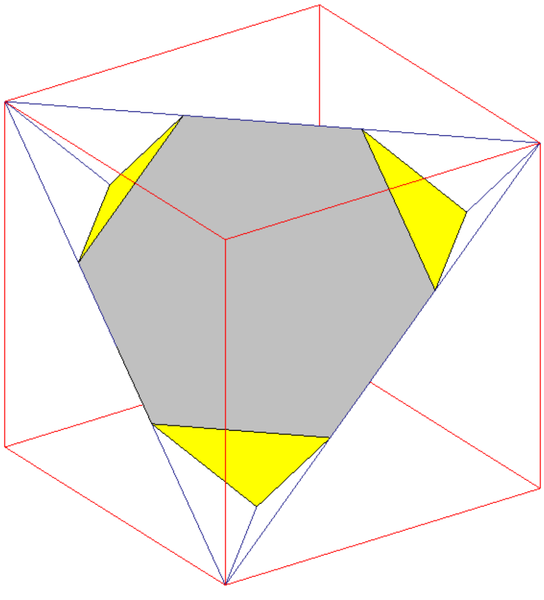
Zeichnung des Tetraederstumpfes:

```
> T1:=plottools[polygon]([a1,b1,c2,c1,b4,a2],style=patch,linest
  yle=1,color=grey):
  T2:=plottools[polygon]([a1,b2,c3,c4,b3,a2],style=patch,linest
  yle=1,color=grey):
  T3:=plottools[polygon]([d1,c2,b1,b2,c3,d2],style=patch,linest
  yle=1,color=grey):
  T4:=plottools[polygon]([d1,c1,b4,b3,c4,d2],style=patch,linest
  yle=1,color=grey):
  T5:=plottools[polygon]([a1,b1,b2],style=patch,linestyle=1,col
  or=yellow):
  T6:=plottools[polygon]([a2,b3,b4],style=patch,linestyle=1,col
  or=yellow):
  T7:=plottools[polygon]([d2,c3,c4],style=patch,linestyle=1,col
  or=yellow):
  T8:=plottools[polygon]([d1,c1,c2],style=patch,linestyle=1,col
  or=yellow):
  display({T1,T2,T3,T4,T5,T6,T7,T8},scaling=constrained);
```



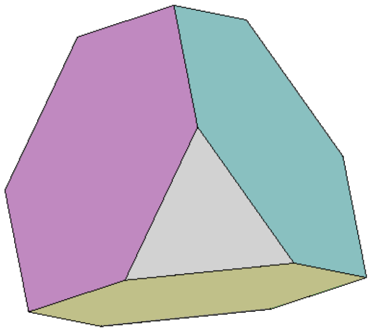
Zeichnung mir Würfel und Tetraeder:

```
> K:=pointplot3d([A4,A2,B1,A4,B3,A2,B3,B1],style=line,color=nav
y,thickness=1):
W:=pointplot3d([A1,A2,A3,A4,A1,B1,B2,A2,B2,B3,A3,B3,B4,A4,B4,
B1],style=line,color=red,thickness=1):
display({T1,T2,T3,T4,T5,T6,T7,T8,K,W},scaling=constrained);
```



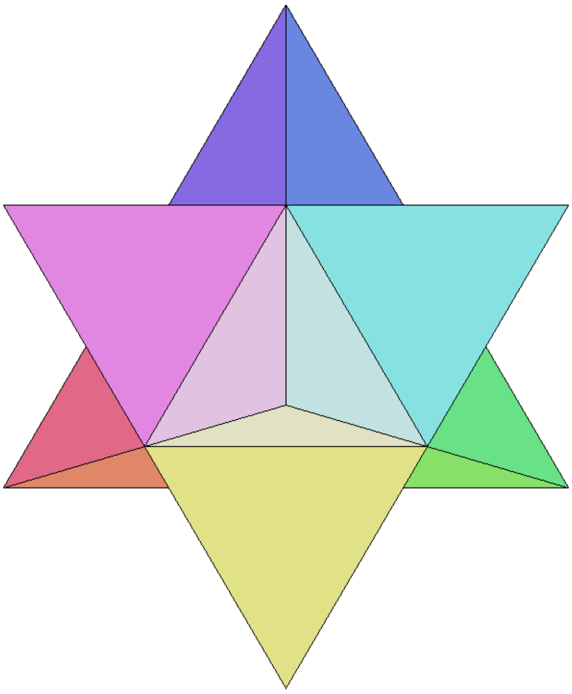
Maple kann es einfacher:

```
> draw(TruncatedTetrahedron(t,point(o,0,0,0),1));
```

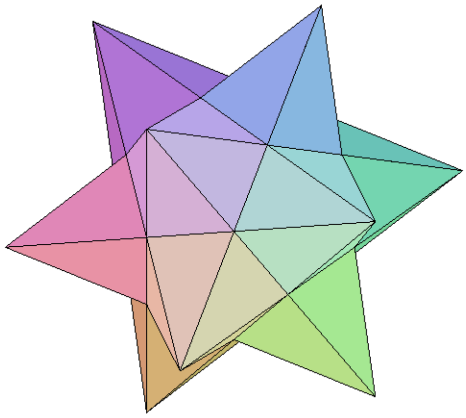


Sterne

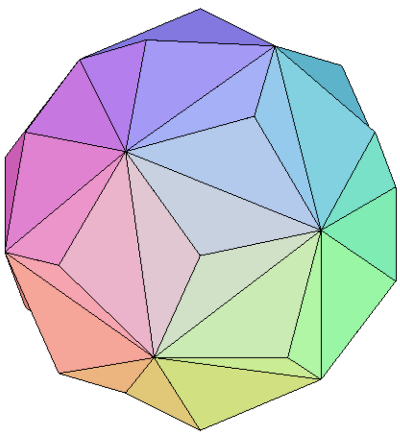
```
> with(geom3d):
> draw(stellate(i1,octahedron(i),1));
```



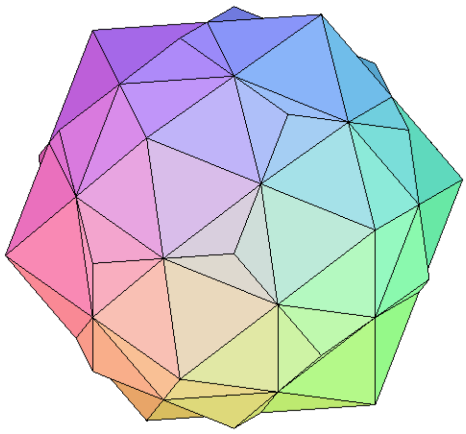
```
> draw(stellate(i1,dodecahedron(i),1));
```



```
> draw(stellate(i1,icosahedron(i),1));
```



```
> draw(stellate(i1,icosidodecahedron(i),1));
```



>

>